

6

Digitale rekenkundige schakelingen

Inhoud

- 6/1 Half adders
- 6/2 Full adders
- 6/3 Magnitude comparatoren
- 6/4 Binaire multipliers
- 6/5 Pariteitsgeneratoren
- 6/6 Arithmetic logic units, look-ahead carry-generatoren en binaire accumulatoren
- 6/7 Diversen

■/■/■ reeds gepubliceerd

■/■/■ gepland voor de volgende aanvullingen

6/2

Full Adders

Inhoud

6/2.1 **Achtergrond-informatie** (aanvulling 28)

6/2.2 **Full Adders 74xx-serie** (aanvulling 4 + 51)

7480	Gated full adder, geïnverteerde in- en uitgangen
7482	2-bits binaire full adder
7483	4-bits binaire full adder met snelle carry
74183	2 x carry-save full adders
74283	4-bits binaire full adder
74385	4 seriële adders/subtractors
74583	4-bits BCD adder met interne carry lookahead

6/2.3 **Full Adders (1)4xxx-serie CMOS** (aanvulling 28)

(1)4008	4-bits full adder met snelle carry
(1)4032	3-voudige seriële adder voor positieve logika
(1)4038	3-voudige seriële adder voor negatieve logika
(1)4560	NBCD adder (subtractor bij gebruik van (1)4561)
(1)4561	9's complement

6/2.1

Achtergrond-informatie

Inleiding

Het is algemeen bekend dat met digitale apparaten, zoals calculators en computers, snel en nauwkeurig kan worden gerekend. Toch zijn de rekenkundige schakelingen die in veel digitale systemen worden gebruikt slechts eenvoudige combinatorische logische circuits (onderling verbonden poorten). In dit gedeelte wordt beschreven hoe met geïntegreerde schakelingen digitaal en binair gecodeerd digitaal (BCD) kan worden opgeteld en afgetrokken.

De digitale schakelingen die hiervoor kunnen worden gebruikt zijn optellers (half- en full adders) en ALU's.

Deze laatste komen in deel 6/6 aan de beurt.

Binair optellen

Voor het optellen van binaire getallen komen slechts vijf basisregels in aanmerking waarvan er vier in figuur 6/2.1-1 te zien zijn. De eerste twee spreken voor zich. De derde regel is, dat binair $1 + 1 = 10$ (decimaal 2).

Sum	Carry out
$0 + 0 = 0$	
$0 + 1 = 1$	
$1 + 0 = 1$	
$1 + 1 = 0$ and carry 1 =	10

Figuur 6/2.1-1: Vier regels voor binair optellen.

Net als bij "gewoon" decimaal optellen moet de 1 in de som naar de volgende kolom worden overgebracht.

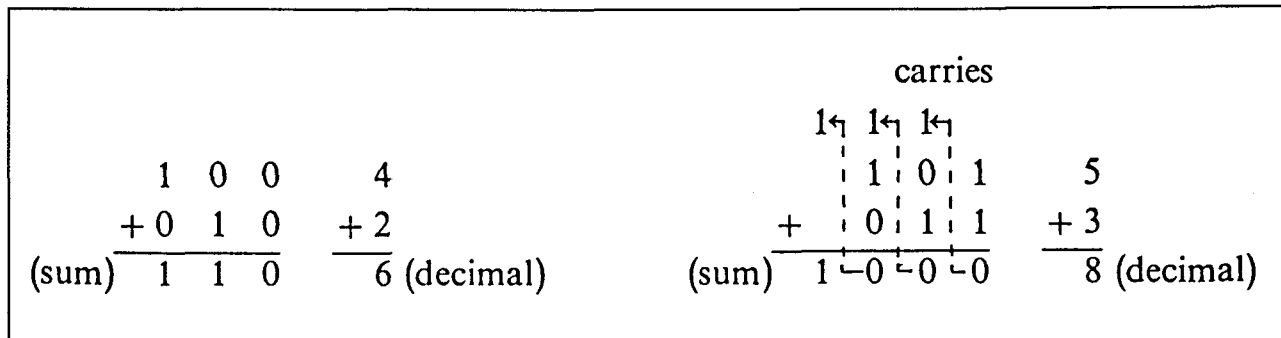
In figuur 6/2.1-2 wordt het gebruik van de basisregels aan de hand van twee eenvoudige voorbeelden toegelicht.

Met deze informatie als basis moet het mogelijk zijn een schakeling te ontwerpen die de optelling kan uitvoeren. Het linker deel van figuur 6/2.1-1 lijkt op een waarheidstabel met twee variabelen. Noemt men de variabelen A en B, de som Σ en de carry-uitgang Co, dan is figuur 6/2.1-3 de bedoelde waarheidstabel. Het "som"-gedeelte hiervan komt overeen met de waarheidstabel van een EXOR-poort en het "carry" gedeelte met die van een AND-poort. In figuur 6/2.1-4 zijn het "bloksymbool" en het schema van de schakeling te zien die aan deze waarheidstabel voldoen.

Aangezien met deze schakeling slechts twee 1-bit getallen kunnen worden opgeteld wordt hij HALF-ADDER (halve opteller) genoemd. Zoals aan de hand van figuur 6/2.1-5 wordt aangetoond kunnen grotere binaire getallen niet met deze schakeling worden opgeteld, maar is daar een FULL-ADDER voor nodig. In de rechtse kolom (1-en) wordt $1 + 1$ opgeteld volgens basisregel 4. De som bedraagt 0 met een carry naar de middelste kolom (2-en).

In deze kolom moet nu $1 + 1 + 1$ worden opgeteld en ontstaat een nieuwe situatie: de som is 11 (decimaal 3).

2.1 Achtergrond-informatie



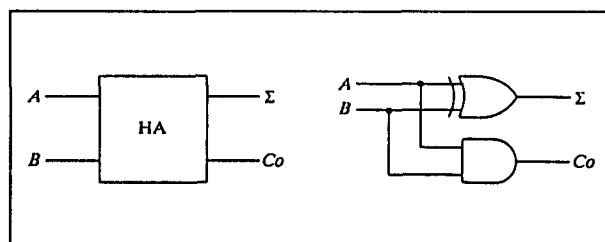
Figuur 6/2.1-2: Twee voorbeelden van binair optellen.

Dit probleem wordt opgelost door de 1 onder de 2-en kolom op de som-positie te plaatsen en een 1 naar de 4-en kolom over te brengen. De 1 bovenaan in de 4-en kolom wordt opgeteld bij twee 0-en en levert dus een 1 op de som-positie op. Het resultaat is 100 (decimaal 4).

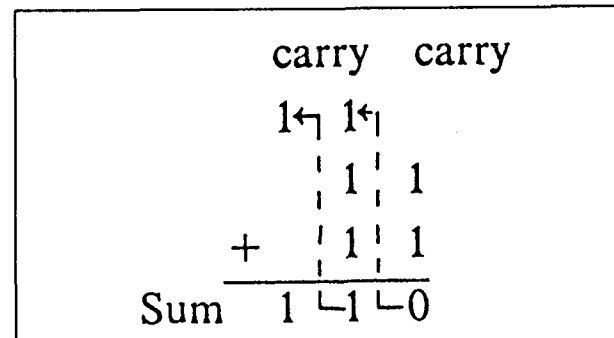
Hiermee is de vijfde basisregel geïntroduceerd die formeel als in figuur 6/2.1-6 wordt geschreven. Merk op dat er nu drie ingangen zijn: A, B en carry-in. De uitgangen zijn Σ en carry-out gebleven. Hiermee wordt aangegeven dat een halve opteller niet werkt als er ook sprake is van een carry-in signaal.

Inputs		Outputs	
A	B	Sum	Carry out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
$A + B$		Σ	Co

Figuur 6/2.1-3: Waarheidstabel van een half-adder.



Figuur 6/2.1-4: Bloksymbool en logisch schema van een half-adder.



Figuur 6/2.1-5: Eenvoudige binaire optelling.

Carry		Carry	
$A + B +$	in	sum	out
1 + 1 +	1	= 1	carry 1 = 11

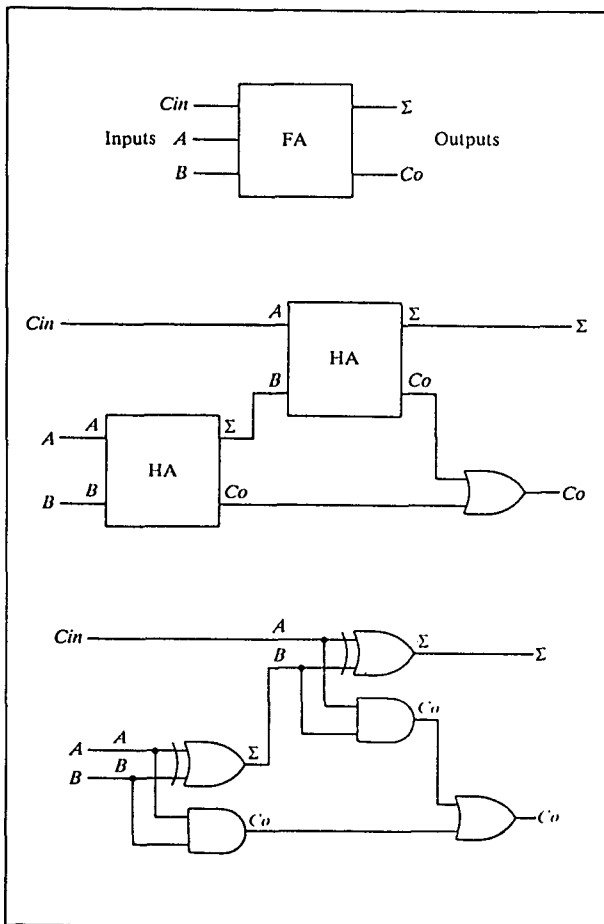
Figuur 6/2.1-6: De vijfde basisregel voor binair optellen.

De optelschakeling met drie ingangen wordt FULL-ADDER genoemd. Figuur 6/2.1-7 toont het bloksymbool, de opbouw met be-

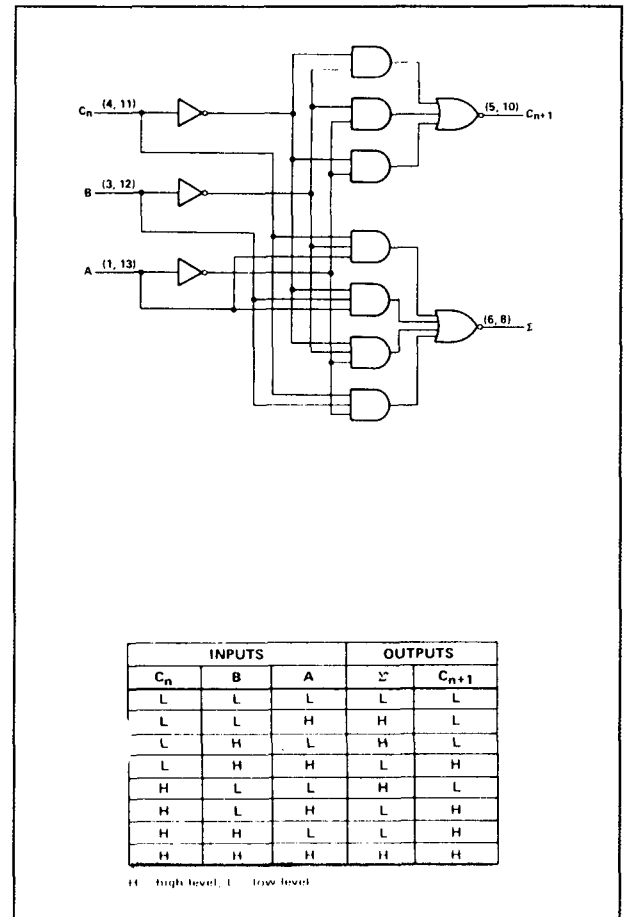
2.1 Achtergrond-informatie

hulp van twee half-adders en een OR-poort en de complete logische schakeling hiervan. De half-adder kan overigens bij het optellen van grotere binaire getallen wel op de plaats van de 1-en worden gebruikt. In de praktijk zal dit echter zelden voorkomen aangezien half-adders niet als geïntegreerde schakeling verkrijgbaar zijn.

Figuur 6/2.1-8 toont tenslotte het functionele schema (positieve logika) en de waarheidstabel van de veel gebruikte full-adder 74LS183.



Figuur 6/2.1-7: Bloksymbool, opbouw met behulp van half-adders en een OR-poort en functioneel schema van een full-adder.



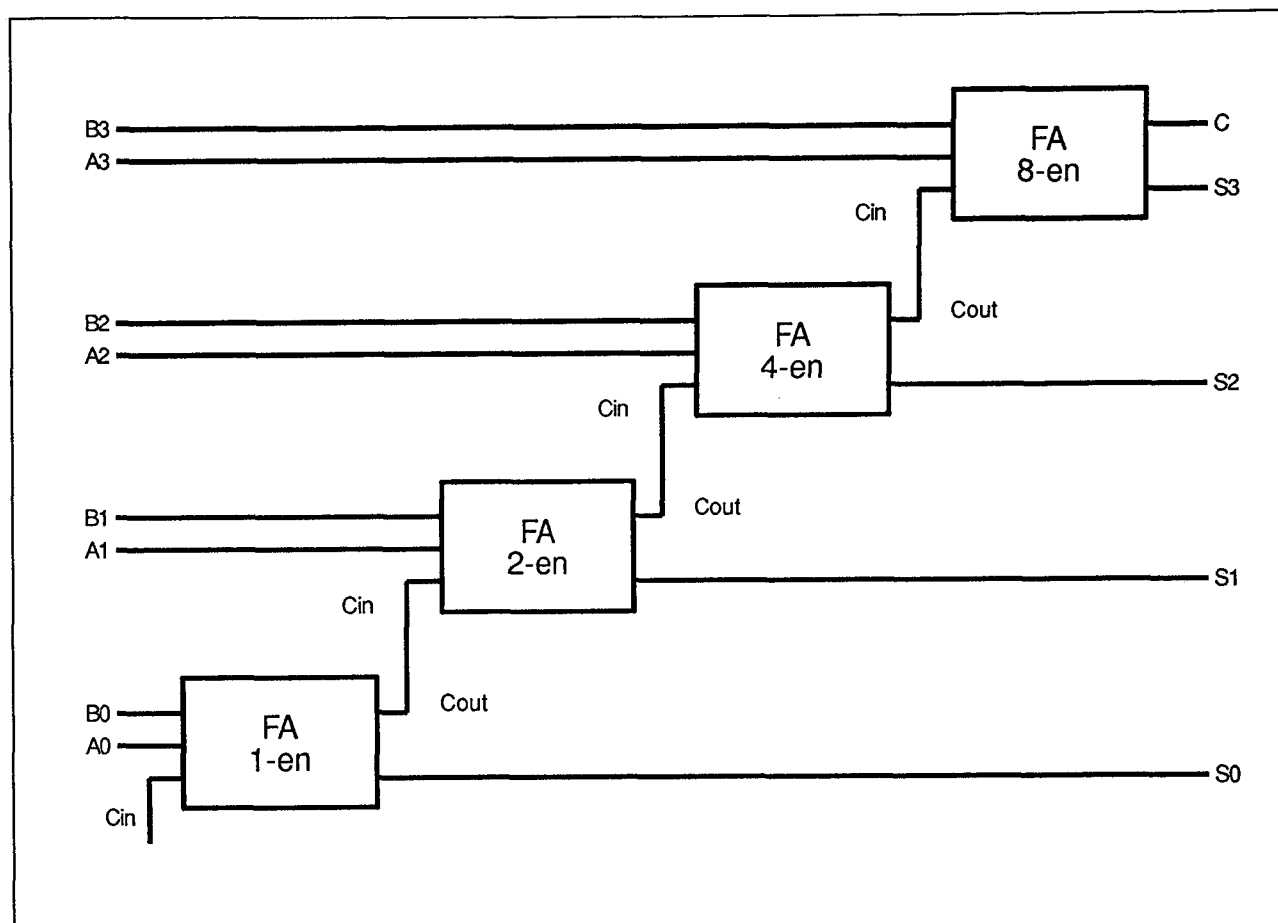
Figuur 6/2.1-8: Functioneel schema en waarheidstabel van één van de optellers in de full-adder 74LS183.

Parallele en seriële optellers

Binair optellen kan op twee manieren gebeuren: parallel of in serie. Bij parallelle optelling worden de binaire getallen tegelijkertijd op de ingangen gezet, waardoor de resulterende som vrijwel direct ontstaat. Bij seriële optelling worden eerst de cijfers van de 1-en kolom opgeteld, dan die van de 2-en kolom plus de carry, de 4-en kolom plus de carry, enzovoorts.

De resultaten moeten telkens in een schuifregister worden opgeslagen. Vooral wanneer grote getallen moeten worden opgeteld heeft de seriële opteller dus nogal wat tijd nodig.

2.1 Achtergrond-informatie



Figuur 6/2.1-9: Schema van een 4-bits parallele opteller.

De parallele opteller

Parallele optellers zijn sneller maar ook gecompliceerder. In figuur 6/2.1-9 is een 4-bit parallele opteller getekend waarbij vier full-adders zijn gebruikt. De onderste full-adder (1-en kolom: A0, B0 en S0) heeft een gearde carry-ingang en zou dus ook door een half-adder vervangen kunnen worden. De volgende full-adder (2-en kolom) telt A1, B1 en de carry-out van de eerste full-adder bij elkaar op.

Ook bij de derde en de vierde full-adder wordt telkens de carry-uitgang van de vorige adder op de carry-ingang van de volgende aangesloten.

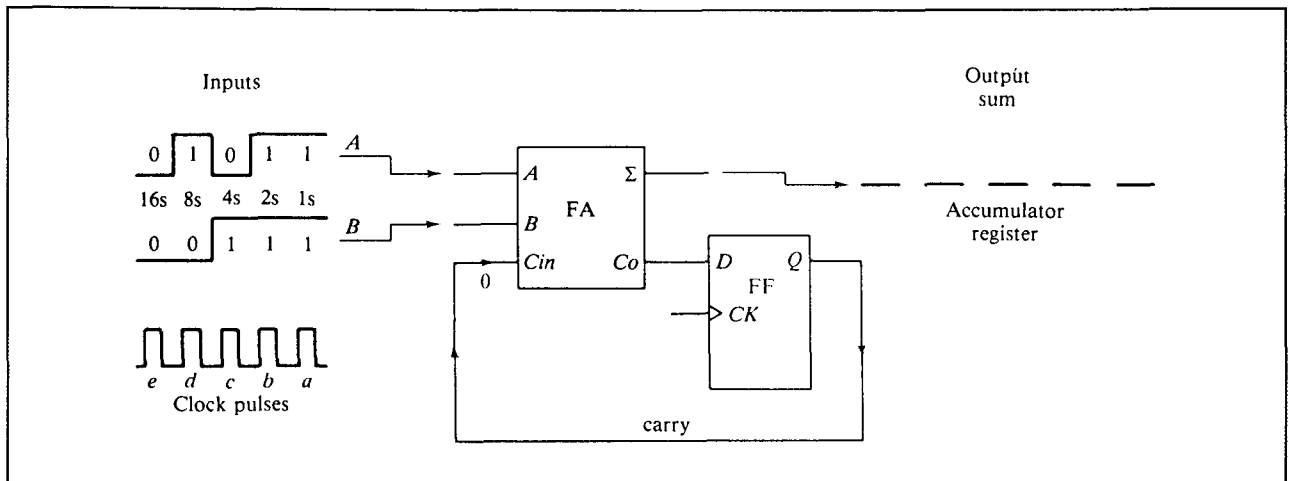
Deze 4-bit opteller heeft zodoende een "ripple-carry". De laatste carry-uitgang is een "overflow" en vormt de 16-en kolom van deze opteller. Wanneer nu bijvoorbeeld $A = 1001$

(decimaal 9) bij $B = 1110$ (decimaal 14) wordt opgeteld verschijnt 10111 (decimaal 23) op de som-uitgangen. Voor het optellen van grotere getallen moeten nog meer full-adders worden gebruikt.

De seriële opteller

Een seriële opteller is minder gecompliceerd, maar wel langzamer dan een parallele. In figuur 6/2.1-10 is de opbouw van een seriële opteller (zonder schuifregisters) te zien. Het belangrijkste onderdeel hiervan is de full-adder FA die telkens twee bits bij elkaar optelt. Eerst de bits van de 1-en, dan die van de 2-en, enzovoorts. De carry wordt telkens met behulp van een D-type flip-flop FF vertraagd, zodat de carry van de 1-en kolom bij de bits van de 2-en kolom kan worden opgeteld, enzovoorts.

2.1 Achtergrond-informatie



Figuur 6/2.1-10: Het hart van een seriële opteller.

In het voorbeeld van figuur 6/2.1-10 wordt binair 01011 opgeteld bij 00111. Beide getallen moeten zijn opgeslagen in schuif-rechts schuifregisters linksboven in dit schema. De som-uitgang van de full-adder moet op zijn beurt weer worden aangesloten op een 5-bit serie-in/schuif-rechts schuifregister. Dit "som-register" (ook wel accumulator genoemd) is eerst leeg. De vijf klokpulsen die aan de linkerzijde zijn getekend, worden tegelijk toegevoerd aan de D flip-flop en aan de A-, B- en som-schuifregisters.

De werking is als volgt. Bij het begin van de optelling zijn $A = 1$, $B = 1$ en $Cin = 0$. Als resultaat hiervan worden $\Sigma = 0$ en $Co = 1$. Na de eerste klokpuls (a) zijn de inhouden van de schuifregisters één plaats naar rechts geschoven. Het som-schuifregister bevat nu het deel 0---- van de som, terwijl een carry van 1 wordt toegevoerd aan de Cin-ingang van de full-adder. Vlak voor klokpuls b zijn $A = 1$, $B = 1$ en $Cin = 1$, waardoor $\Sigma = 1$ en $Co = 1$ worden. Ook na de tweede klokpuls (b) zijn alle inhouden weer één plaats naar rechts geschoven.

Het som-register bevat nu het deel 10--- van de som, terwijl $A = 0$, $B = 1$ en $Cin = 1$ zijn. Vlak voor de laatste klokpuls (e) zijn de ingangen $A = 0$, $B = 0$ en $Cin = 1$, zodat na deze klokpuls de inhoud van het som-register 10010 is geworden.

Binair aftrekken

Het aftrekken van twee binaire getallen kan op twee manieren gebeuren. Met half- en full-subtractors (die niet als geïntegreerde schakeling verkrijgbaar zijn) of door full-adders te gebruiken als aftrekkers. Eerst worden de officiële subtractors behandeld en daarna de methode met de meer gebruikelijke (iets) gewijzigde adders.

Figuur 6/2.1-11 laat de basisregels voor het binaire aftrekken van twee bits zien. Bij een aftrekking wordt het bovenste getal het afgetrektal genoemd en het onderste de aftrekker. Het resultaat is natuurlijk het verschil.

0	-	0	=	0	
0	-	1	=	1	and borrow 1
1	-	0	=	1	
1	-	1	=	0	

Figuur 6/2.1-11: Vier regels voor binair aftrekken.

Regel 1 is duidelijk. Bij regel 2 wordt 1 afgetrokken van het kleinere getal 0. Dit is alleen mogelijk als van de kolom aan de linkerzijde een 1 "geleend" kan worden. De regels 3 en 4 leveren verder geen problemen op.

2.1 Achtergrond-informatie

Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0
$A - B$		Di	Bo

Figuur 6/2.1-12: Waarheidstabel van een halve aftrekker (half-subtractor).

Net als bij het optellen lijken deze vier basisregels op een waarheidstabel.

In figuur 6/2.1-12 zijn zij omgezet in de waarheidstabel voor een half-subtractor. Wanneer men naar de kolom van het verschil ($Di = \text{difference}$) kijkt, dan blijkt dat Di (net als Σ bij de optellers) overeenkomt met een EXOR-functie. De borrow-kolom Bo heeft als vergelijking $\bar{A} \cdot B = Y$ zodat die met behulp van een inverter en een AND-poort verwezenlijkt kan worden.

Het bloksymbool en het logische schema van de half-subtractor dat volgens bovenstaande waarheidstabel kan worden opgezet, zijn te zien in figuur 6/2.1-13. Wanneer het schema van de half-subtractor wordt vergeleken met dat van de half-adder (figuur 6/2.1-4), dan blijkt het enige verschil de inverter aan de A-ingang van de AND-poort van de half-subtractor te zijn.

In het rekenvoorbeeld van figuur 6/2.1-14 ziet men dat meerdere leningen (borrows) nodig zijn.

Voor de aftrekking in de 1-en kolom kan een half-subtractor worden toegepast (of een full-subtractor met geaarde borrow-ingang Bin). Voor de aftrekkingen in de andere kolommen (2-en, 4-en, 8-en, 16-en en 32-en) moeten in elk geval full-subtractors worden gebruikt.

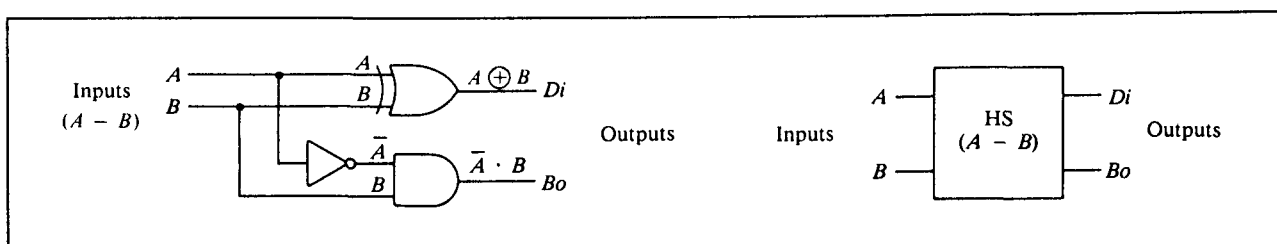
Het bloksymbool, de opbouw met behulp van half-subtractors en het logische schema van een dergelijke full-subtractor zijn te zien in figuur 6/2.1-15. De ingangen zijn A (aftrektal), B (aftrekker) en Bin (borrow ingang), terwijl Di (verschil) en Bo (borrow-uit) de uitgangen zijn.

Om te kunnen volgen tot hoever geleend wordt, moet Bin telkens met Bo van de lagere aftrekker worden verbonden. Figuur 6/2.1-16 toont tenslotte de waarheidstabel van een full-subtractor.

Parallele en seriële aftekkers

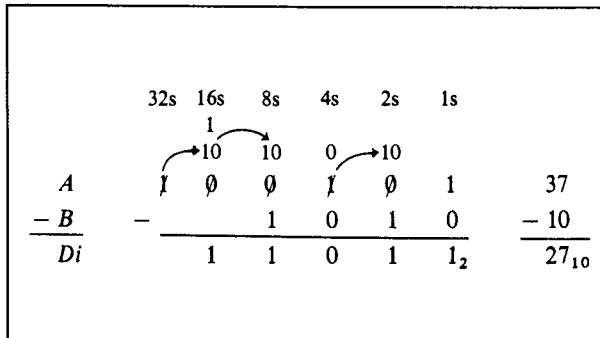
Net als binair optellen kan ook aftrekken op twee manieren gebeuren, namelijk parallel of in serie.

Bij parallel aftrekken worden de binaire getallen tegelijkertijd op de ingangen gezet, waardoor het verschil bijna direct ontstaat. Bij seriële aftrekken worden eerst de cijfers van de 1-en kolom afgetrokken en daarna die van de 2-en kolom minus de carry, de 4-en kolom minus de carry, enzovoorts. De resultaten moeten achtereenvolgens in een schuifregister worden opgeslagen, waarvoor dus telkens een klokpuls nodig is.



Figuur 6/2.1-13: Logisch schema en bloksymbool van een half-subtractor.

2.1 Achtergrond-informatie



Figuur 6/2.1-14: Rekenvoorbeeld van een aftrekking waarin borrows voorkomen.

De parallelle aftrekker

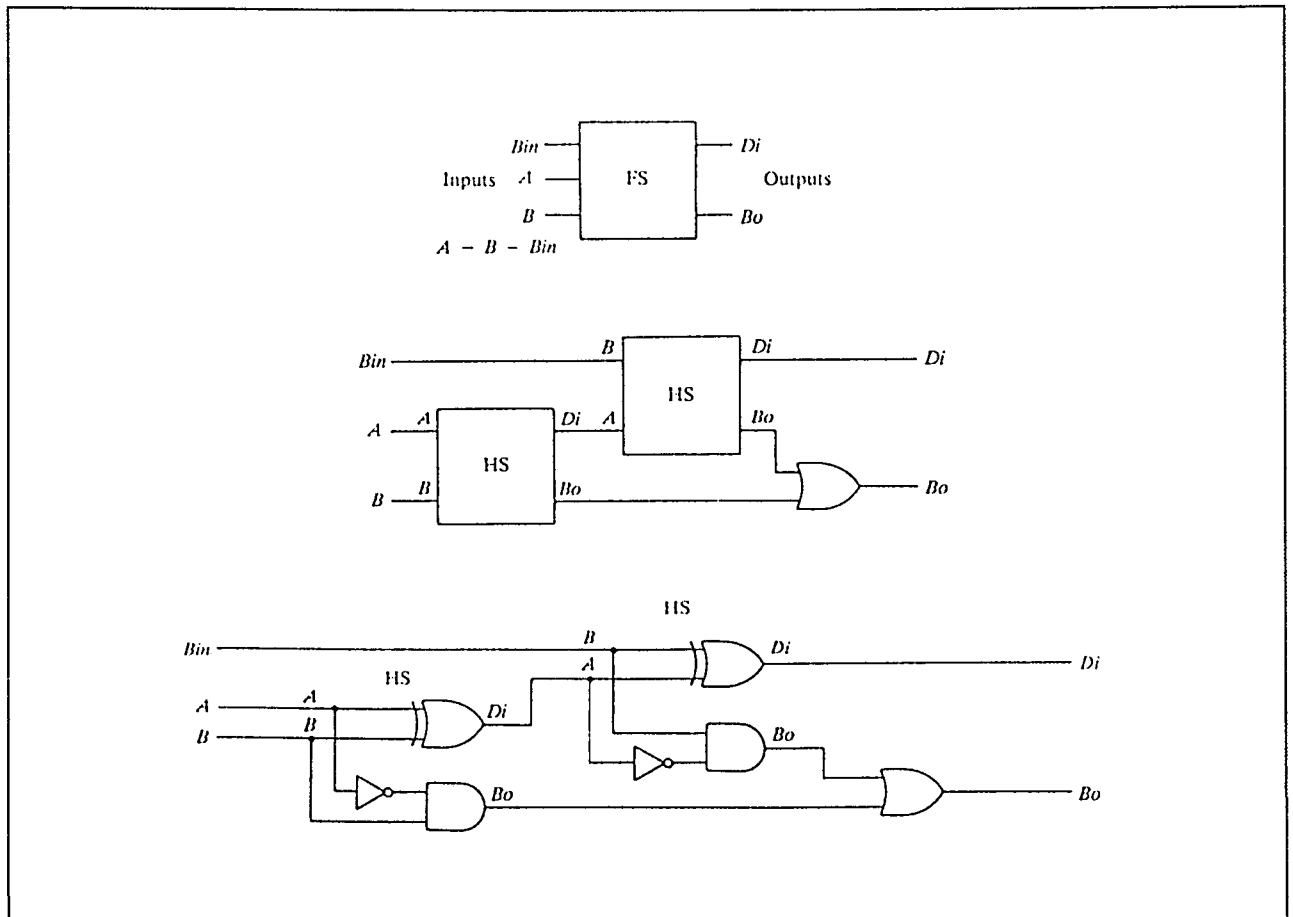
Parallele aftrekkers zijn veel sneller, maar ook gecompliceerder. In figuur 6/2.1-17 toont een 4-bit parallelle aftrekker die uit vier full-

subtractors (FS) is samengesteld. De onderste full-subtractor (1-en kolom: A0, B0 en Di0) heeft een geaarde borrow-ingang en mag dus ook door een half-subtractor vervangen worden. De volgende full-subtractor (2-en kolom) trekt B1 en de borrow-out van de eerste subtractor van A1 af.

Ook bij de derde en de vierde full-subtractor wordt telkens de borrow-uitgang van de vorige subtractor op de borrow-ingang van de volgende aangesloten. Deze 4-bit aftrekker heeft dus een "ripple-borrow".

Wanneer nu bijvoorbeeld B = 1001 (decimaal 9) van A = 1110 (decimaal 14) wordt afgetrokken verschijnt 0101 (decimaal 5) op de verschil-uitgangen.

Voor grotere getallen moeten natuurlijk meer full-subtractors worden gebruikt.



Figuur 6/2.1-15: Bloksymbool, opbouw met behulp van half-subtractors en logisch schema van een volledige aftrekker (full-subtractor).

2.1 Achtergrond-informatie

Inputs			Outputs	
(A)	(B)	Borrow in (Bin)	Borrow out (Bo)	
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1
$A - B - Bin$			Di	Bo

Figuur 6/2.1-16: Waarheidstabel van een full-subtractor.

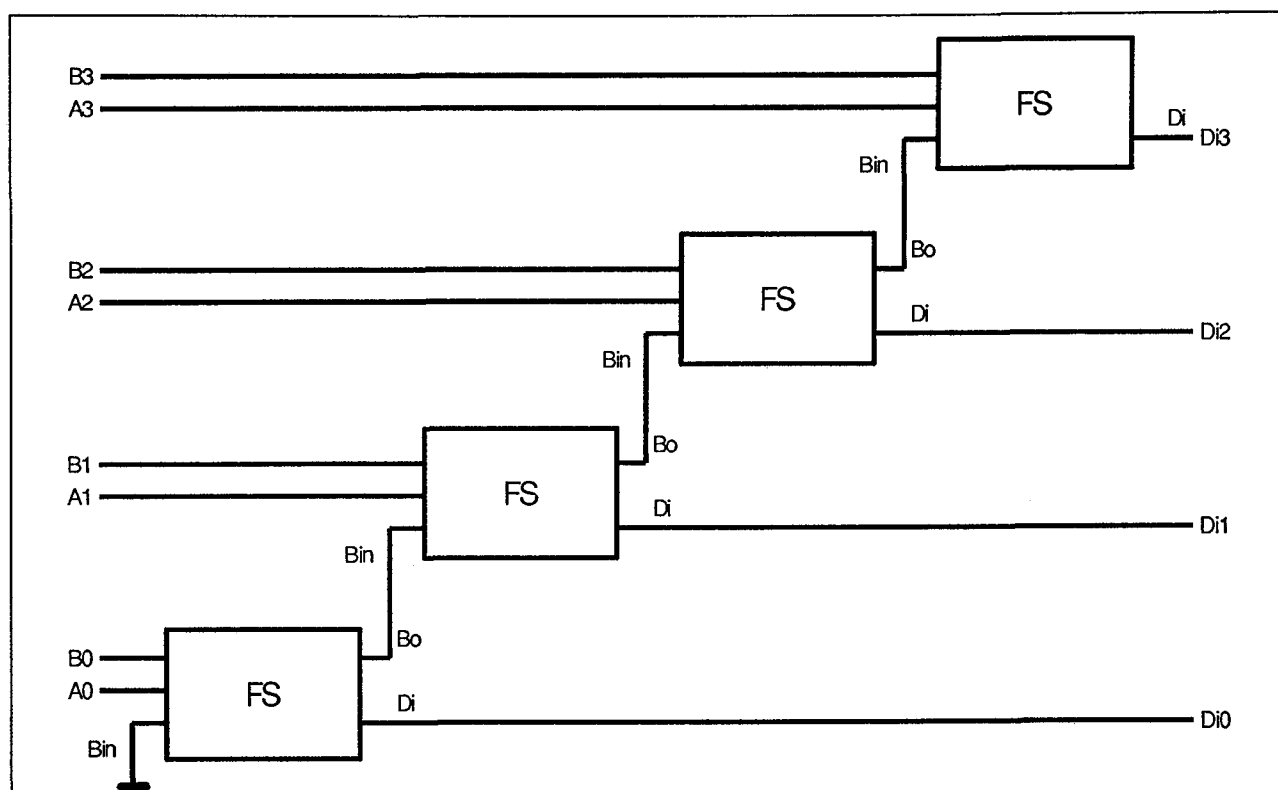
De seriële aftrekker

Figuur 6/2.1-18 toont het schema van een seriële aftrekker (full-subtractor) waarbij de

borrow-signalen apart worden toegevoerd. Volgens de waarheidstabel van figuur 6/2.1-16 wordt de Di-uitgang achtereenvolgens 0, 1, 1, 0, 1, 0, 0 en 1 op de tijdstippen a tot en met h. De Bo-uitgang krijgt op deze tijdstippen de waarden 0, 1, 0, 0, 1, 0, 1 en 1. De borrow-uitgang kan op dezelfde manier als bij een seriële opteller via een flip-flop op de borrow-ingang worden aangesloten. Voor het aan- en afvoeren van de seriële informatie worden ook hierbij schuifregisters gebruikt.

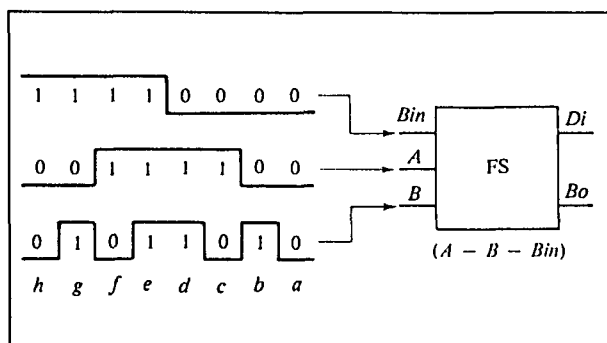
Volledigheidshalve is in figuur 6/2.1-19 nog het complete schema van een seriële opteller/aftrekker opgenomen.

Hierbij wordt de full-adder dus zowel voor optellen als voor aftrekken gebruikt, terwijl ook het B-schuifregister twee taken heeft, namelijk opslag van X en daarna van het resultaat.



Figuur 6/2.1-17: Schema van een 4-bits parallelle aftrekker.

2.1 Achtergrond-informatie



Figuur 6/2.1-18: Een seriële full-subtractor met aparte data- en borrow-signalen (zie ook tekst).

Aftrekken met behulp van adders

In figuur 6/2.1-19 werd al vooruit gelopen op het gebruik van adders voor aftrekken. Omdat optellen en aftrekken in berekeningen bijna even vaak voorkomen, kan hiermee het totale aantal benodigde schakelingen flink worden beperkt.

Om het aftrekken terug te brengen tot optellen kan gebruik worden gemaakt van de "two's complement" of van de "one's complement" notatie. Bij beide notaties geldt dat het hoogste bit van een getal het teken aan- geeft (0 = positief, 1 = negatief).

Het two's complement wordt gevormd door:

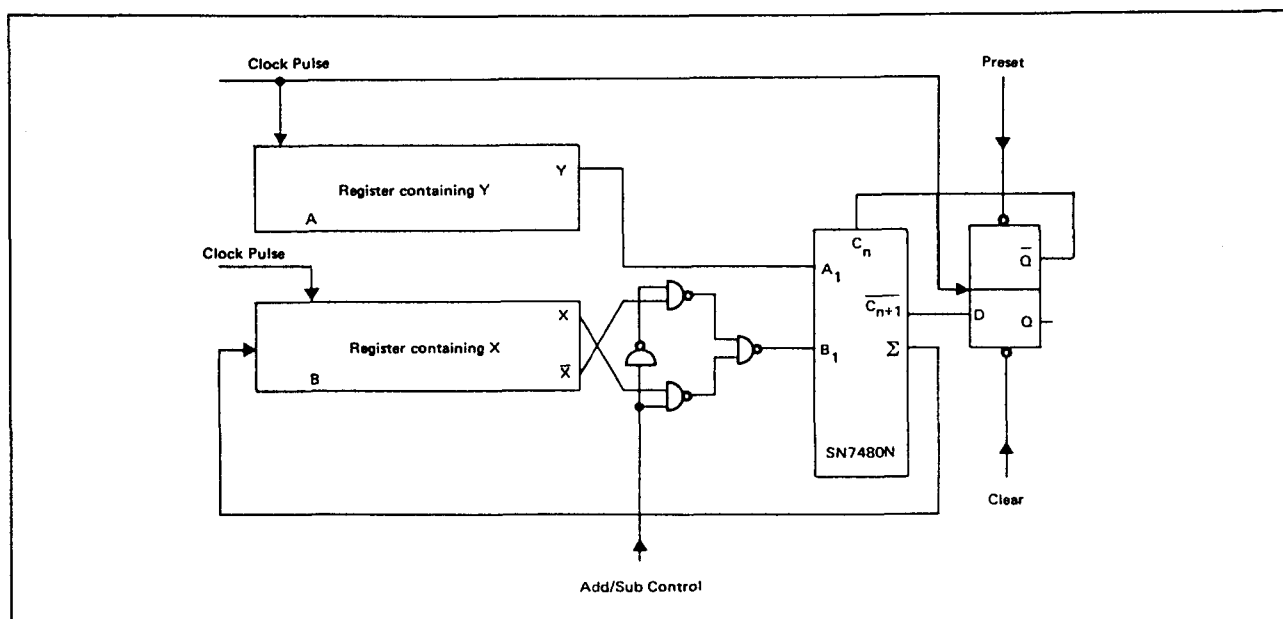
- het getal in binaire vorm te schrijven;
- alle bits te inverteren (= one's complement);
- hierbij één op te tellen.

Om bijvoorbeeld in 4-bits notatie het getal -7 te verkrijgen moet van de binaire waarde (0111) het complement (1000) plus 1 worden genomen, dus (1001).

Op dezelfde manier wordt -1 in 8-bits notatie 1111 1111, wordt -4 = 1111 0000 en -129 = 0111 1111.

Wanneer met behulp van het two's complement wordt afgetrokken (is dus het optellen van het geconverteerde getal) moet de carry worden genegeerd.

Figuur 6/2.1-20 geeft hiervan enkele reken- voorbeelden, terwijl figuur 6/2.1-21 laat zien hoe dit voor twee 2-bits getallen wordt uitge- voerd.



Figuur 6/2.1-19: Een complete seriële opteller/af trekker, waarbij gebruik wordt gemaakt van de full-adder 7480.

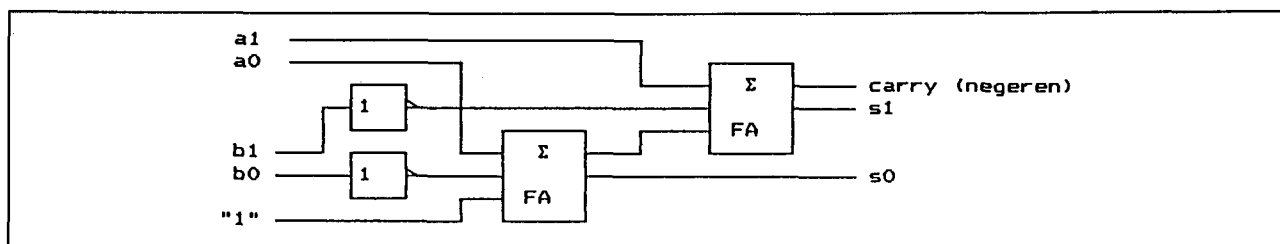
2.1 Achtergrond-informatie

4	0100	-4	1100
-2	1110	+2	0010
<hr/>			
2	10010	-2	1110
negeren ↑			

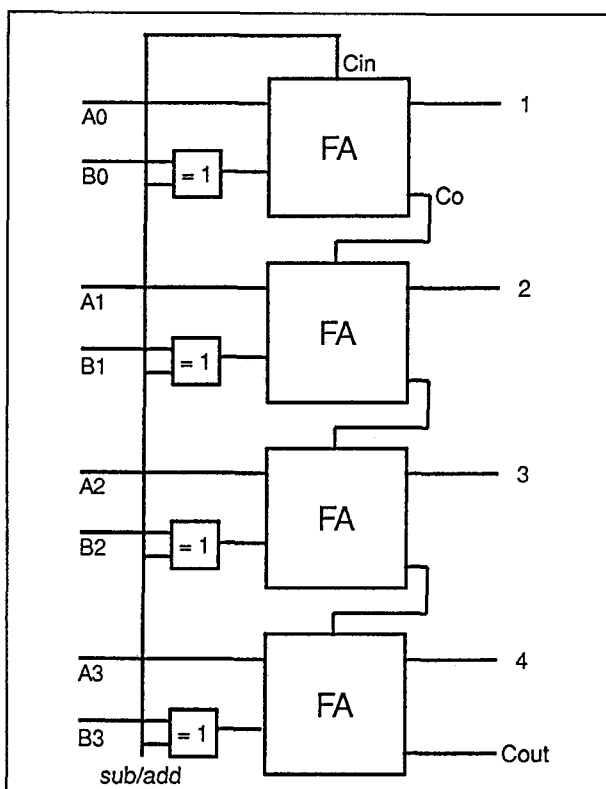
Figuur 6/2.1-20: Enkele rekenvoorbeelden van aftrekken met gebruik van het two's complement.

Tenslotte laat figuur 6/2.1-22 een schema zien waarmee naar wens kan worden opgeteld of afgetrokken.

Bij het optellen ($\text{sub/add} = 0$) wordt geen van de bits geïnverteerd en is ook de extra optelling van één niet nodig. In dit geval wordt het carry-bit gebruikt. Bij het aftrekken ($\text{sub/add} = 1$) worden de B-bits geïnverteerd, terwijl bij het resultaat één wordt opgeteld. Het carry-bit wordt nu genegeerd.



Figuur 6/2.1-21: Een schakeling waarmee twee 2-bits getallen volgens de two's complement methode kunnen worden afgetrokken.



Figuur 6/2.1-22: 4-bits optel/aftrekschakeling met full-adders voor de two's complement methode.

Bij het aftrekken met behulp van het one's complement wordt het end-around carry-bit bij het resultaat opgeteld.

Deze methode is wat eleganter dan de vorige:

- neem het complement van de binaire notatie van de aftrekker;
- tel het aftrektal hierbij op;
- tel de carry van de belangrijkste plaats op bij de 1-en plaats.

Figuur 6/2.1-23 geeft een rekenvoorbeeld ($13 - 4 = 9$) van deze methode.

De 9 blijft ongewijzigd, van de 4 wordt het complement genomen en bij de 9 opgeteld, waarna ook de carry bij het geheel wordt opgeteld.

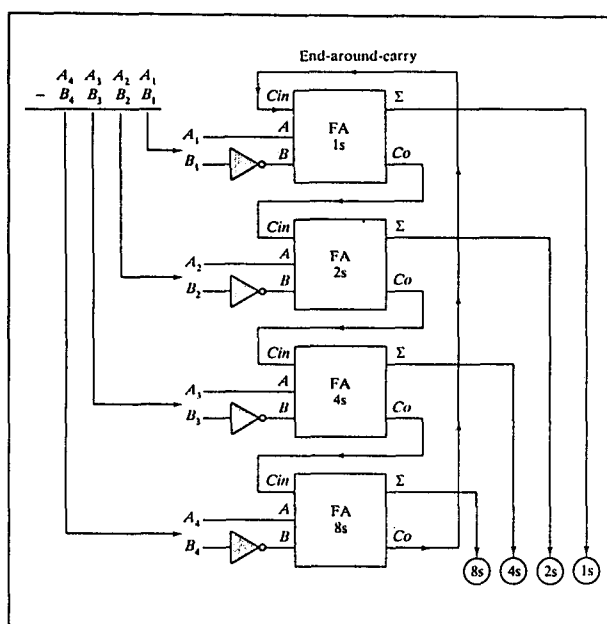
Deze aftrekking kan met de schakeling van figuur 6/2.1-24 worden uitgevoerd. Het verschil van dit schema met het aftrekgedeelte van figuur 6/2.1-22 zit hem in de carry die in dit geval naar de laagste full-adder wordt teruggeleid.

Bij optellen moeten de inverters vervallen en moet de verbinding van de end-around carry worden verbroken.

2.1 Achtergrond-informatie

Binary subtraction		Is complement and end-around-carry subtraction	
Minuend	1101		1101
Subtrahend	- 0100	complement subtrahend	+ 1011
Difference	1001	and add	+ 1000
			end-around-carry + 1
			1001
			Difference

Figuur 6/2.1-23: Rekenvoorbeeld van de one's complement methode.



Figuur 6/2.1-24: Een 4-bit aftrekschakeling waarbij full-adders volgens de one's complement methode worden toegepast.

2.1 Achtergrond-informatie

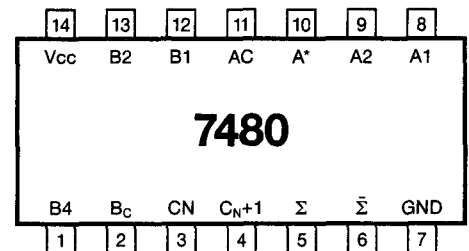
6/2.2

Full Adders 74xx-serie

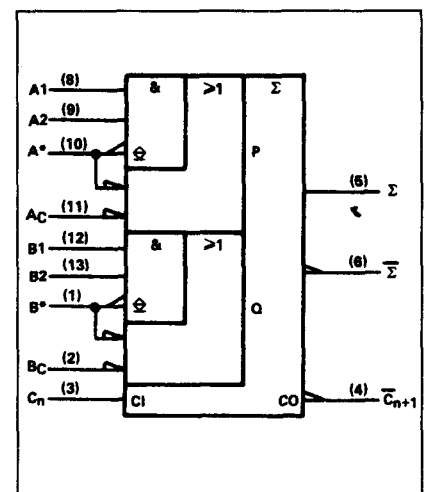
7480

Gated full adder,
geinverteerde in- en uitgangen

Figuur 6/2-80.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}	21									mA
I _{os}	-18 -57									mA
T _{plh} ¹⁾	13									ns
T _{phl} ¹⁾	8									ns
T _{plh} ²⁾	18									ns
T _{phl} ²⁾	38									ns
T _{plh} ³⁾	52									ns
T _{phl} ³⁾	62									ns
T _{plh} ⁴⁾	38									ns
T _{phl} ⁴⁾	56									ns



INPUTS			OUTPUTS		
C _n	B	A	C _{n+1}	Σ̄	Σ
L	L	L	H	H	L
L	L	H	H	L	H
L	H	L	H	L	H
L	H	H	L	H	L
H	L	L	H	L	H
H	L	H	L	H	L
H	H	L	L	H	L
H	H	H	L	L	H

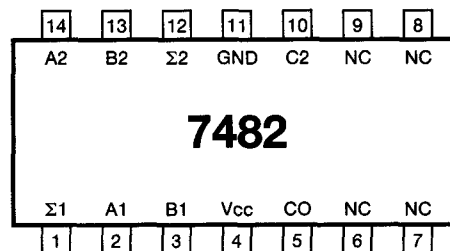
1) C_N to C_N + 1 2) B_C to C_N + 1 3) A_C to Σ 4) B_C to Σ

2.2 74xx-serie

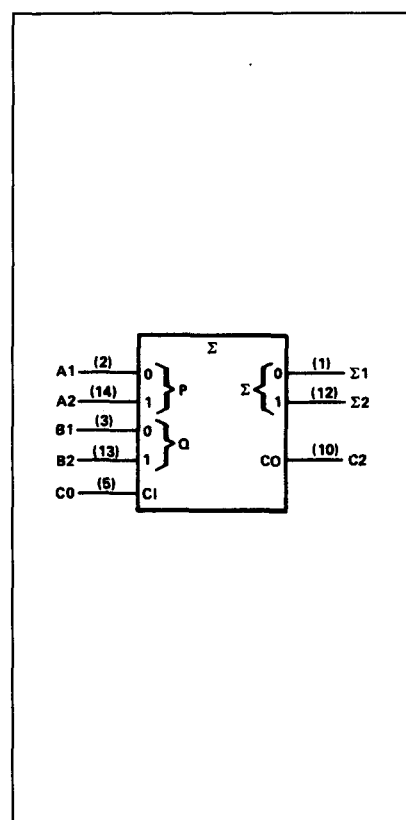
7482

2-bits
binaire full adder

Figuur 6/2-82.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}	35 -70									mA
I _{os}	-18 -70									mA
T _{plh} ¹⁾ _{max}	34									ns
T _{phl} ¹⁾ _{max}	40									ns
T _{plh} ²⁾ _{max}	40									ns
T _{phl} ²⁾ _{max}	35									ns
T _{plh} ³⁾ _{max}	38									ns
T _{phl} ³⁾ _{max}	42									ns
T _{plh} ⁴⁾ _{typ} max	12 19									ns
T _{phl} ⁴⁾ _{typ} max	17 27									ns



1) CO to Σ1 2) B2 to Σ2 3) CO to Σ2 4) CO to C2

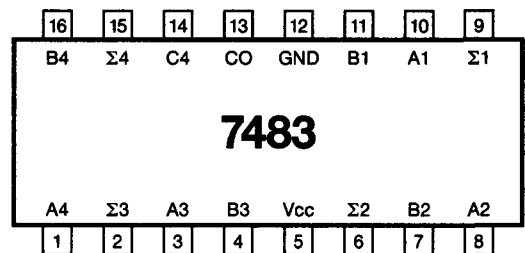
INPUTS				OUTPUTS			
A1	B1	A2	B2	WHEN CO = L		WHEN CO = H	
				Σ1	Σ2	C2	Σ1
L	L	L	L	L	L	L	H
H	L	L	L	H	L	L	L
L	H	L	L	H	L	L	H
H	H	L	L	L	H	L	H
L	L	H	L	L	H	L	H
H	L	H	L	H	H	L	L
L	H	H	L	H	H	L	L
H	H	H	L	L	L	H	H
L	L	L	H	L	H	L	H
H	L	L	H	H	H	L	L
L	H	L	H	H	H	L	L
H	H	L	H	L	L	H	H
L	L	H	H	L	L	H	H
H	L	H	H	H	L	H	L
L	H	H	H	H	L	H	L
H	H	H	H	L	H	H	H

2.2 74xx-serie

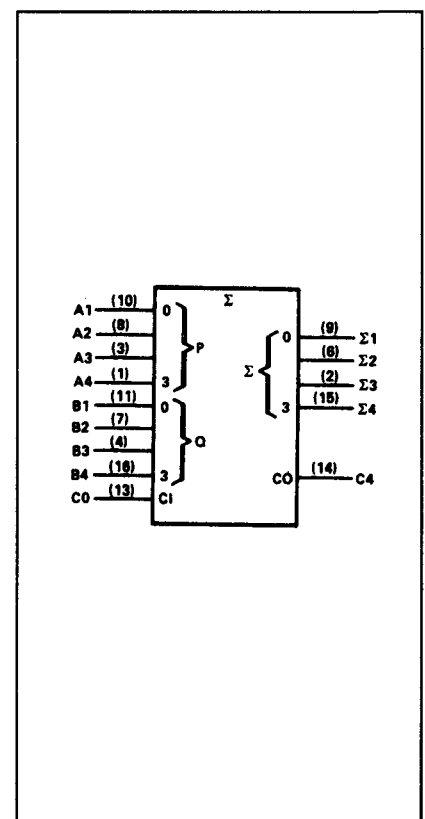
7483

4-bits binaire full adder,
met snelle carry

Figuur 6/2-83.



LOGICA	TTL ⁵⁾	L	F	S	LS ⁶⁾	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc} dis ^H _L	66				22 19 19			0.05 ⁵⁾		mA
I _{os}	-18 -70				-20 -100					mA
T _{plh} ¹⁾	14				16			350		ns
T _{phl} ¹⁾	12				15			350		ns
T _{plh} ²⁾	16				15			300		ns
T _{phl} ²⁾	16				15			300		ns
T _{plh} ³⁾	9				11			120		ns
T _{phl} ³⁾	11				11			120		ns
T _{plh} ⁴⁾	9				11			250		ns
T _{phl} ⁴⁾	11				12			250		ns



1) CO to Any Σ 2) A_i or B_i to Σ_i 3) CO to C₄ 4) A_i or B_i to C₄ 5) 7483 A 6) 74LS 83A
7) μA

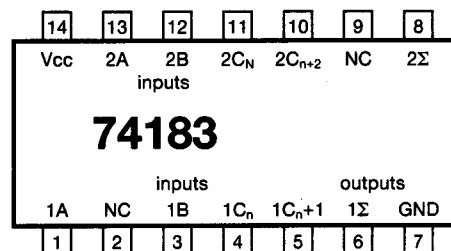
INPUT				OUTPUT							
				WHEN CO = L				WHEN CO = H			
A1	B1	A2	B2	Σ1	Σ2	Σ3	Σ4	Σ1	Σ2	Σ3	Σ4
L	L	L	L	L	L	L	L	H	L	L	L
L	L	L	H	L	L	L	L	L	L	L	L
L	L	L	L	L	L	L	L	L	L	L	L
L	L	L	H	L	L	L	L	L	L	L	L
L	L	H	L	L	L	L	L	L	L	L	L
L	L	H	H	L	L	L	L	L	L	L	L
L	H	L	L	L	L	L	L	L	L	L	L
L	H	L	H	L	L	L	L	L	L	L	L
L	H	H	L	L	L	L	L	L	L	L	L
L	H	H	H	L	L	L	L	L	L	L	L
H	L	L	L	L	L	L	L	L	L	L	L
H	L	L	H	L	L	L	L	L	L	L	L
H	L	H	L	L	L	L	L	L	L	L	L
H	L	H	H	L	L	L	L	L	L	L	L
H	H	L	L	L	L	L	L	L	L	L	L
H	H	L	H	L	L	L	L	L	L	L	L
H	H	H	L	L	L	L	L	L	L	L	L
H	H	H	H	L	L	L	L	L	L	L	L

2.2 74xx-serie

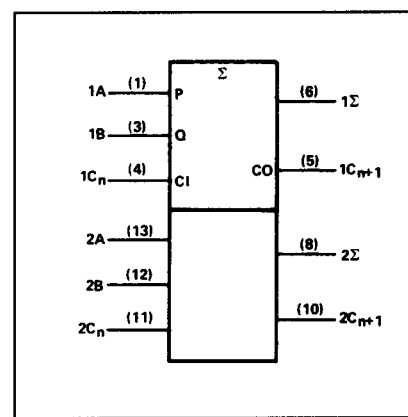
74183

2 carry-save full adders

Figuur 6/2-183.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
Icc	H				8					mA
	L				10					
Ios					-20					mA
					-100					
Tplh					9					ns
Tphl					20					ns

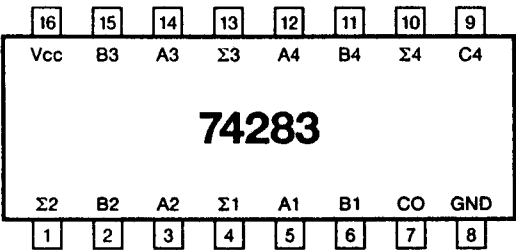


INPUTS			OUTPUTS	
C _n	B	A	Σ	C _{n+1}
L	L	L	L	L
L	L	H	H	L
L	H	L	H	L
L	H	H	L	H
H	L	L	H	L
H	L	H	L	H
H	H	L	L	H
H	H	H	H	H

2.2 74xx-serie

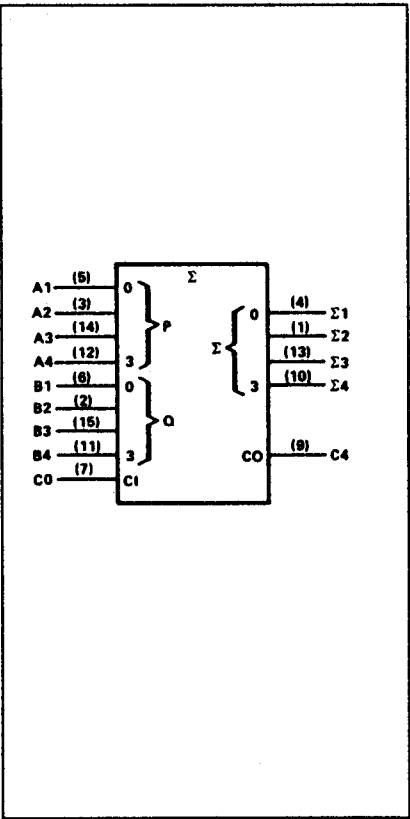
74283

4-bits binaire full adder



Figuur 6/2-283.

LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABLE PARAMETERS									Eenheid
I _{cc}	H L	66 56			80 95	19 19			80 ⁵⁾	mA
I _{os}		-18 -70			-20 -100	-20 -100				mA
T _{plh} ¹⁾		14			11	16			23	ns
T _{phl} ¹⁾		12			12	15			23	ns
T _{phl} ²⁾		16			12	15			23	ns
T _{phl} ²⁾		16			11.5	15			23	ns
T _{phl} ³⁾		9			6	11			17	ns
T _{phl} ³⁾		11			7.5	11			17	ns
T _{phl} ⁴⁾		9			7.5	11			17	ns
T _{phl} ⁴⁾		11			8.5	12			17	ns



1) CO to Any Σ 2) A_i or B_i to Σ_i 3) CO to C₄ 4) A_i or B_i to C₄ 5) μA

INPUT				OUTPUT							
				WHEN CO = L				WHEN CO = H			
				WHEN C ₃ = L	WHEN C ₃ = H	WHEN C ₂ = L	WHEN C ₂ = H	WHEN C ₃ = L	WHEN C ₃ = H	WHEN C ₂ = L	WHEN C ₂ = H
A ₁	B ₁	A ₂	B ₂	Σ ₁	Σ ₂	Σ ₃	Σ ₄	Σ ₁	Σ ₂	Σ ₃	Σ ₄
L	L	L	L	L	L	L	L	H	L	L	L
L	L	L	H	L	L	L	L	H	L	L	L
L	L	H	L	L	L	L	L	H	L	L	L
L	L	H	H	L	L	L	L	H	L	L	L
L	H	L	L	L	L	L	L	H	L	L	L
L	H	L	H	L	L	L	L	H	L	L	L
L	H	H	L	L	L	L	L	H	L	L	L
L	H	H	H	L	L	L	L	H	L	L	L
H	L	L	L	L	L	L	L	H	L	L	L
H	L	L	H	L	L	L	L	H	L	L	L
H	L	H	L	L	L	L	L	H	L	L	L
H	L	H	H	L	L	L	L	H	L	L	L
H	H	L	L	L	L	L	L	H	L	L	L
H	H	L	H	L	L	L	L	H	L	L	L
H	H	H	L	L	L	L	L	H	L	L	L
H	H	H	H	L	L	L	L	H	L	L	L

2.2 74xx-serie

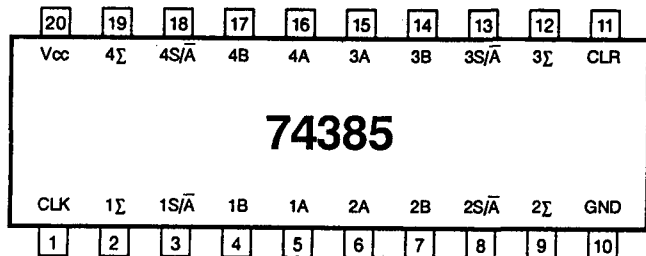
74385

4 seriële

adders/subtractors

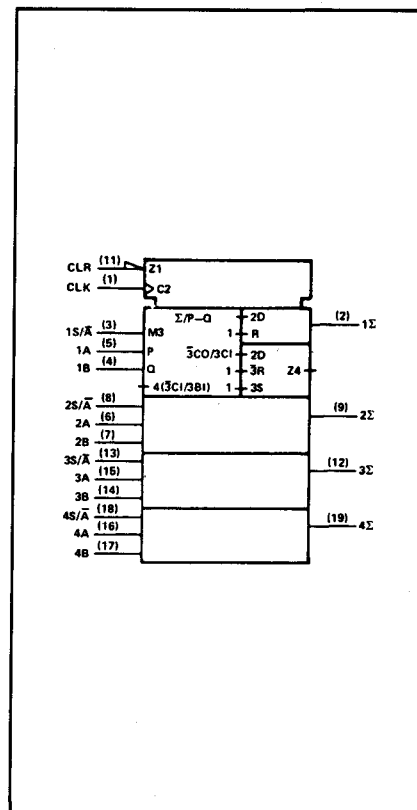
(onafhankelijke Two's-complement optellers/aftrekkers)

Figuur 6/2.2-385.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}			68		48					mA
I _{os}			-60		-20					mA
			-150		-100					
T _{plh} ¹⁾			6		14					ns
T _{phl} ¹⁾			7		18					ns
T _{phl} ²⁾			9		18					ns
f _{max}			100		40					MHz

¹⁾ CLK → Σ
²⁾ Clear → Σ

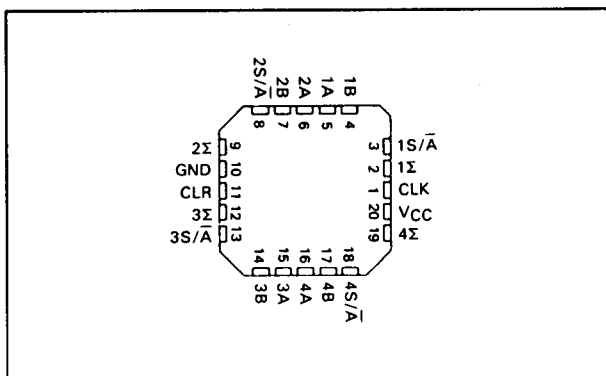


2.2 74xx-serie

SELECTED FUNCTION	INPUTS					DATA IN CARRY FLIP-FLOP		Σ OUTPUT
	CLR	S/A	A	B	CLK	BEFORE †	AFTER †	AFTER †
Clear	L	L	X	X	X	L	L	L
	L	H	X	X	X	H	H	L
Aod	H	L	L	L	↑	L	L	L
	H	L	L	L	↑	H	L	H
	H	L	L	H	↑	L	L	H
	H	L	L	H	↑	H	L	L
	H	L	H	L	↑	L	H	L
	H	L	H	L	↑	H	H	L
	H	L	H	H	↑	L	H	L
	H	L	H	H	↑	H	H	H
Subtract	H	H	L	L	↑	L	L	H
	H	H	L	L	↑	H	L	L
	H	H	L	H	↑	L	L	L
	H	H	L	H	↑	H	L	H
	H	H	H	L	↑	L	H	L
	H	H	H	L	↑	H	H	H
	H	H	H	H	↑	L	L	H
	H	H	H	H	↑	H	H	L

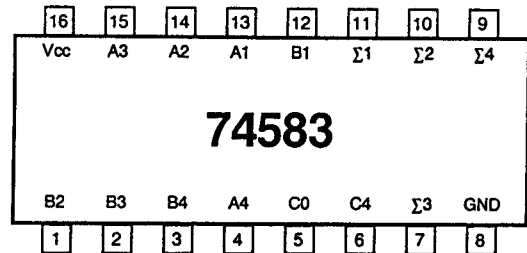
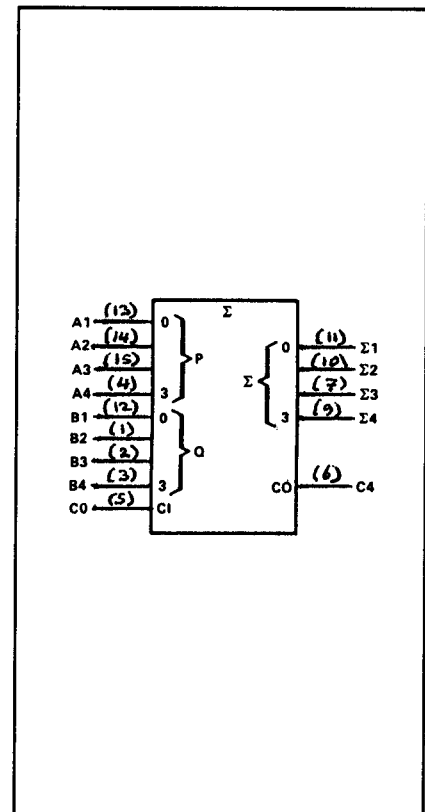
H = high level, L = low level, X = irrelevant,
 † = transition from low to high level at the clock input

Waarheidstabel.



Ceramische chip-carrier.

4-bits BCD adder met interne carry lookahead

[illegible]

- 1) A of B $\rightarrow \Sigma$
- 2) C0 \rightarrow C4

3) $A \text{ of } B \rightarrow C4$

[illegible]

Waarheidstabel.

6/2.3

Full Adders (1)4xxx-serie CMOS

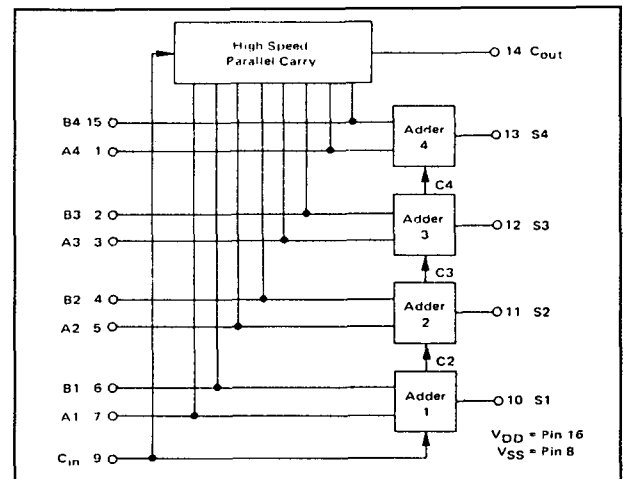
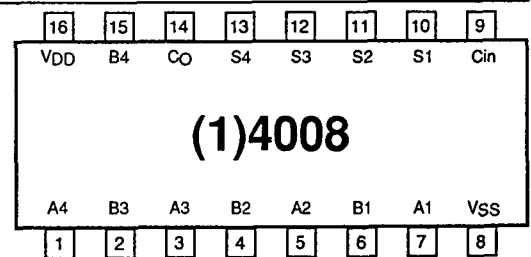
(1)4008

4-bits full adder met snelle carry

Figuur 6/2.3-08

C _{in}	B	A	C _{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Waarheidstabel.



Functioneel blokschema.

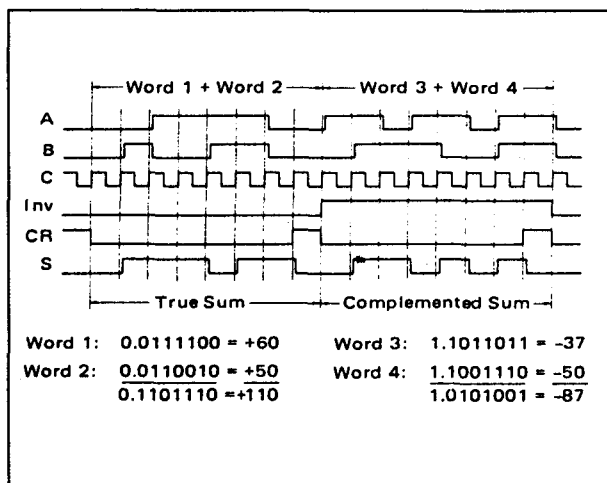
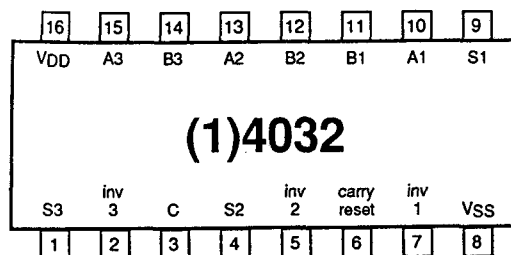
o.a. leverbaar:
 MC 14008 A/B/C, CD 4008 A/B,
 HEF 4008 B

2.3 Full Adders (1)4xxx-serie CMOS

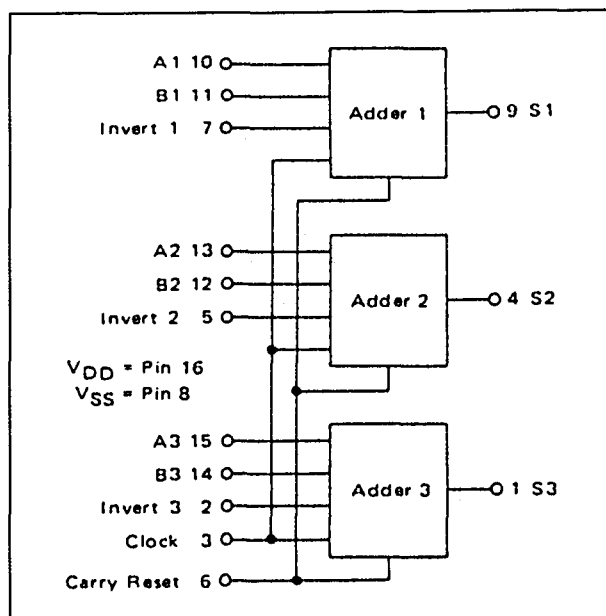
(1)4032

3-voudige seriële adder
voor positieve logica

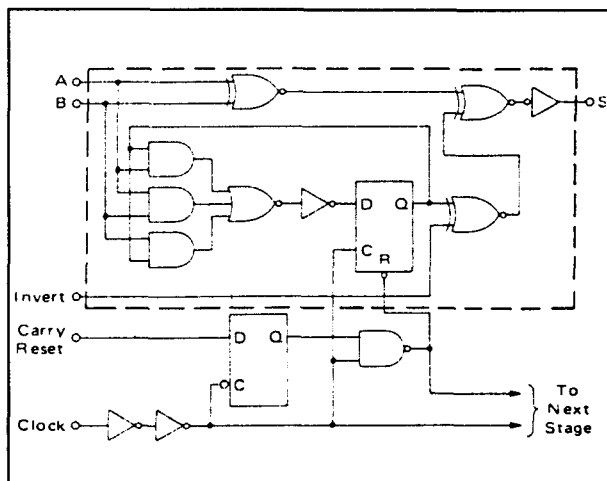
Figuur 6/2.3-32



Timing.



Functioneel blokschema.



Logisch schema van één sectie.

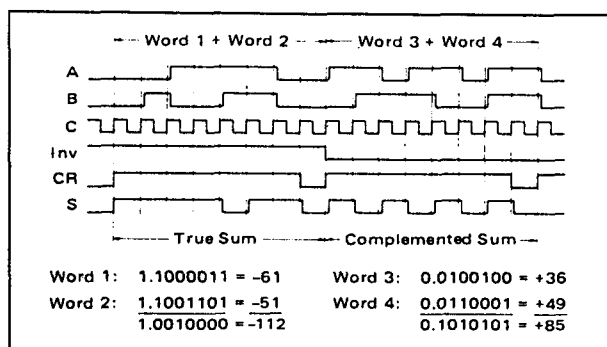
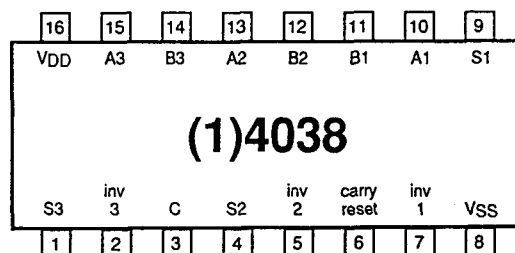
o.a. leverbaar:
MC 14032 A/B/C, CD 4032 A/B,
HD 14032 B

2.3 Full Adders (1)4xxx-serie CMOS

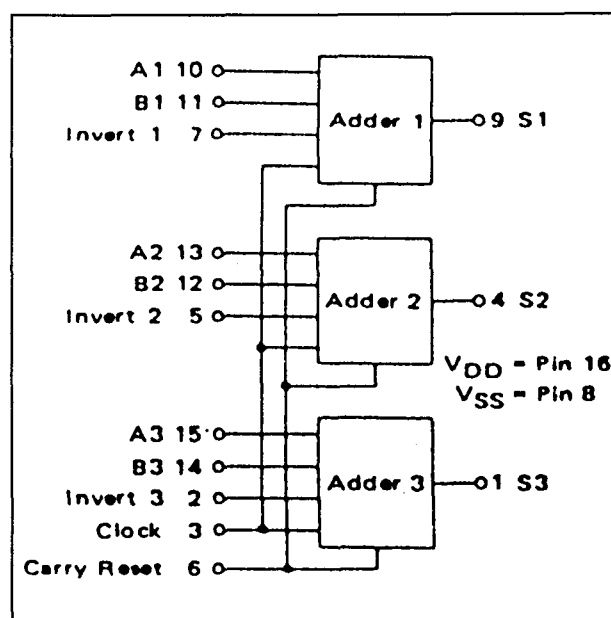
(1)4038

3-voudige seriële adder
voor negatieve logika

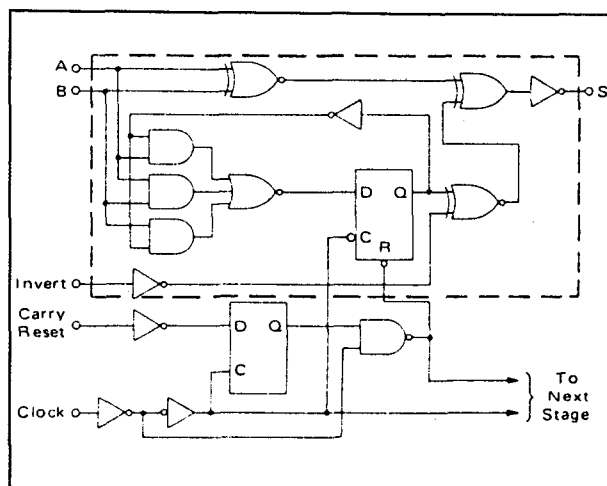
Figuur 6/2.3-38



Timing.



Functioneel blokschema.



Logisch schema van één sectie.

o.a. leverbaar:
MC 14038 A/B/C, CD 4038 A/B, MSM 4038

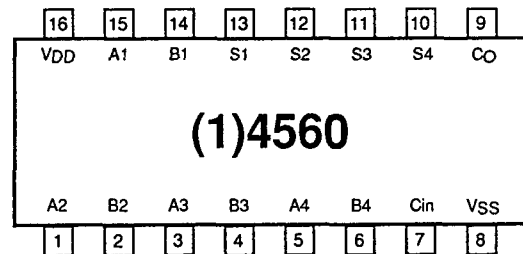
2.3 Full Adders (1)4xxx-serie CMOS

(1)4560

NBCD adder
(natural binary coded
decimal)

kan ook gebruikt worden voor aftrek-
ken als één set ingangen m.b.v. de
(1)4561 wordt gecomplementeerd

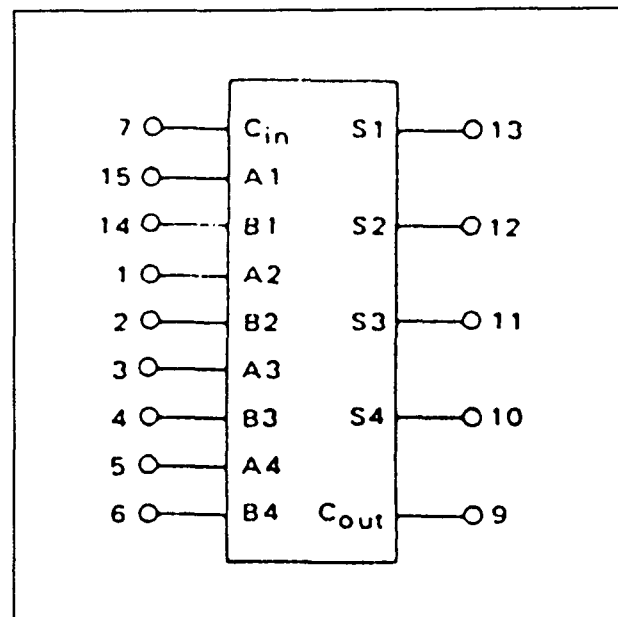
Figuur 6/2.3-560



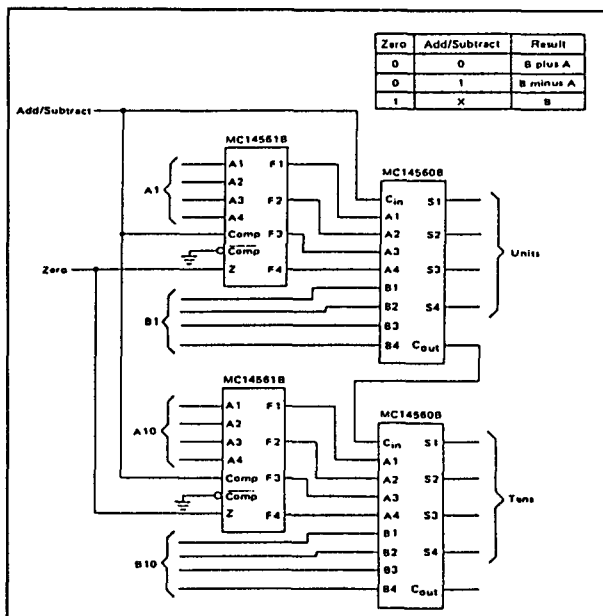
INPUT										OUTPUT			
A4	A3	A2	A1	B4	B3	B2	B1	C _{in}	C _{out}	S4	S3	S2	S1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	0	1	1	0	0	1	1	1
0	0	1	0	0	0	0	1	1	1	0	1	0	0
0	1	1	1	0	0	0	0	0	1	0	0	0	1
0	1	1	1	0	0	0	1	0	1	0	0	1	0
0	1	0	0	0	1	0	1	0	1	0	0	1	1
0	1	1	0	1	0	0	0	0	1	0	1	0	0
0	1	0	0	1	1	0	1	1	1	1	0	0	1

*Partial truth table to show logic operation for representative input values

Waarheidstabel.



Logisch symbool.



Toepassingsvoorbeeld: parallel optellen/af-trekken.

o.a. leverbaar:
MC 14560 B, TC 4560 B, μ PD 4560B

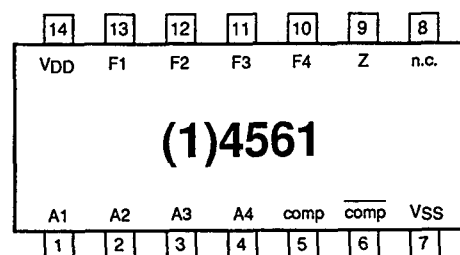
2.3 Full Adders (1)4xxx-serie CMOS

(1)4561

9's complementer

wordt samen met (1)4560 gebruikt
om BCD-getallen af te trekken

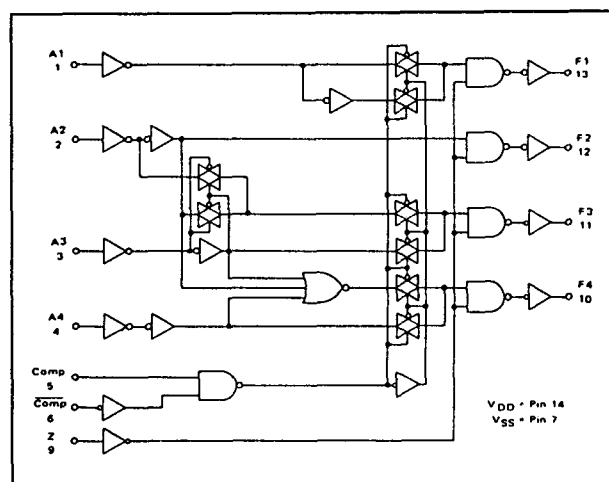
Figuur 6/2.3-561



Z	Comp	Comp	F1	F2	F3	F4	Mode
0	0	0					
0	0	1	A1	A2	A3	A4	Straight-through
0	1	1					
0	1	0	$\bar{A}1$	A2	$A2\bar{A}3 + \bar{A}2A3$	$\bar{A}2\bar{A}3\bar{A}4$	Complement
1	X	X	0	0	0	0	Zero

X = Don't Care.

Waarheidstabel.



Functioneel schema.

	Decimal Equivalent Input	Inputs				Decimal Equivalent Output	Outputs			
		A4	A3	A2	A1		F4	F3	F2	F1
	0	0	0	0	0	9	1	0	0	1
	1	0	0	0	1	8	1	0	0	0
	2	0	0	1	0	7	0	1	1	1
	3	0	0	1	1	6	0	1	1	0
	4	0	1	0	0	5	0	1	0	1
	5	0	1	0	1	4	0	1	0	0
	6	0	1	1	0	3	0	0	1	1
	7	0	1	1	1	2	0	0	1	0
	8	1	0	0	0	1	0	0	0	1
	9	1	0	0	1	0	0	0	0	0
Illegal BCD Input Codes	10	1	0	1	0	7	0	1	1	1
	11	1	0	1	1	6	0	1	1	0
	12	1	1	0	0	5	0	1	0	1
	13	1	1	0	1	4	0	1	0	0
	14	1	1	1	0	3	0	0	1	1
	15	1	1	1	1	2	0	0	1	0

Waarheidstabel (complement-mode).

O.a. leverbaar:

MC 14561 B, HD14561 B, μ PD 4561 B

2.3 Full Adders (1)4xxx-serie CMOS

6/3

Magnitude comparatoren

Inhoud

6/3.1 **Achtergrond-informatie**
(aanvulling 27)6/3.2 **Magnitude comparatoren 74xx-serie TTL en HC**
(aanvulling 48)

7485	4 bit magnitude comparator
74518	8 bit identiteitscomparator, 20 k Ω optrekweerstanden, open-collector uitgang
74519	8 bit identiteitscomparator, open-collector uitgang
74520	8 bit identiteitscomparator, 20 k Ω optrekweerstanden, totempaal uitgang
74521	8 bit identiteitscomparator, totempaal uitgang
74522	8 bit identiteitscomparator, 20 k Ω optrekweerstanden, open-collector uitgang
74526	16 bit fuse-programmable identiteitscomparator
74527	8 bit fuse-programmable identiteitscomparator en 4 bit comparator
74528	12 bit fuse-programmable identiteitscomparator
74677	16 bit adrescomparator met enable
74678	16 bit adrescomparator met latch
74679	12 bit adrescomparator met enable
74680	12 bit adrescomparator met latch
74682	8 bit magnitude comparator, 20 k Ω optrekweerstanden
74683	8 bit magnitude comparator, 20 k Ω optrekweerstanden, open-collector uitgang
74684	8 bit magnitude comparator
74685	8 bit magnitude comparator, open-collector uit
74686	8 bit magnitude comparator
74687	8 bit magnitude comparator, open-collector uit
74688	8 bit identiteitscomparator met totempaal uitgang
74689	8 bit identiteitscomparator met open-collector uitgang
74866	8 bit magnitude comparator
74885	8 bit magnitude comparator, P-ingangen latchbaar, diverse logische- of wiskundige vergelijkingen

- 6/3.3 Magnitude comparatoren (1)4xxx-serie CMOS**
(aanvulling 48)
(1)4063 4 bit magnitude comparator
(1)4585 4 bit magnitude comparator

6/3.1

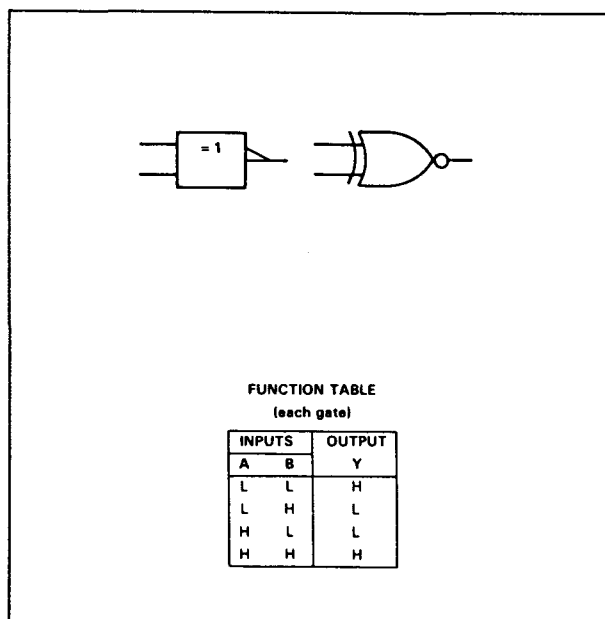
Achtergrond-informatie

Inleiding

Comparatoren worden gebruikt om twee binaire woorden met elkaar te vergelijken. De eenvoudigste melden alleen wanneer deze woorden gelijk zijn, terwijl er ook typen zijn die "kleiner dan" en "groter dan" aangeven. Afhankelijk van het gebruik worden ze magnitude (grootte) comparator, adres comparator of identiteits comparator genoemd.

De EXNOR-poort

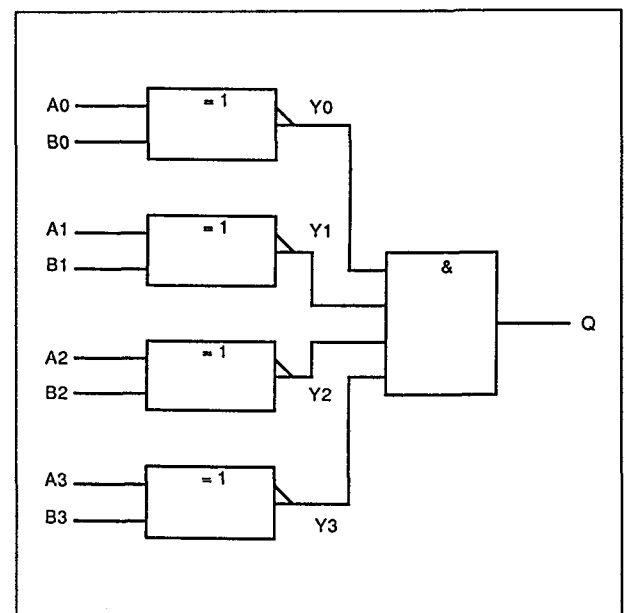
De eenvoudigste vorm van een binaire comparator is de Exclusive-NOR poort, waar de 74810 er bijvoorbeeld vier van heeft.



Figuur 6/3.1-1: Logisch symbool, alternatief logisch symbool en waarheidstabel van één van de EXNOR-poorten in de 74810.

De EXNOR-poort geeft immers aan wanneer twee 1-bits getallen A en B aan elkaar gelijk zijn (zowel beide 1 als beide 0). In figuur 6/3.1-1 zijn hiervan het logische symbool en de waarheidstabel opgenomen.

Het is hiermee wel mogelijk te signaleren dat A ongelijk is aan B, maar niet dat $A > B$ (A groter dan B) of $A < B$ (A kleiner dan B) is.



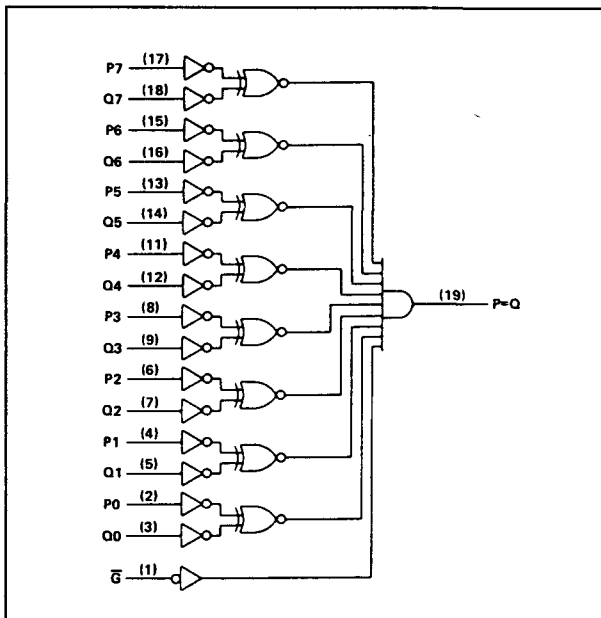
Figuur 6/3.1-2: Schema voor het vergelijken van twee 4-bits woorden (Q = HOOG als $A = B$).

Meervoudig vergelijken

Met een aantal EXNOR-poorten parallel en een AND-poort kunnen grotere binaire getallen (woorden) met elkaar worden vergeleken. Figuur 6/3.1-2 geeft een voorbeeld voor

3.1 Achtergrond-informatie

het vergelijken van twee 4-bit woorden. Wanneer het A-woord gelijk is aan het B-woord zijn alle vier Y-uitgangen van de EXNOR's HOOG, zodat ook de uitgang Q van de AND-poort HOOG gaat. In alle andere gevallen is Q LAAG.



Figuur 6/3.1-3: Functioneel blokschema (positiële logika) van de 8-bit comparator 74ALS518.

Dat dergelijke oplossingen ook werkelijk worden toegepast is te zien in figuur 6/3.1-3 waar het functionele blokschema van de 8-bit identiteits comparator 74ALS518 wordt getoond. Ook hier is te zien dat de uitgang $P = Q$ alleen HOOG wordt als het P-woord gelijk is aan het Q-woord.

A gelijk aan, groter dan of kleiner dan B

Dat voor het detecteren van $A > B$, $A = B$ en $A < B$ een veel ingewikkelder schakeling nodig is, zal duidelijk zijn. In figuur 6/3.1-4 en tabel 6/3.1-1 worden het functionele blokschema en de waarheidstabel van de 4-bit comparator 7485 getoond. Deze schakeling is zelfs nog iets ingewikkelder door de cascade-ingangen waarmee de toestand van naburige comparatoren kan worden doorgegeven.

Toepassingen

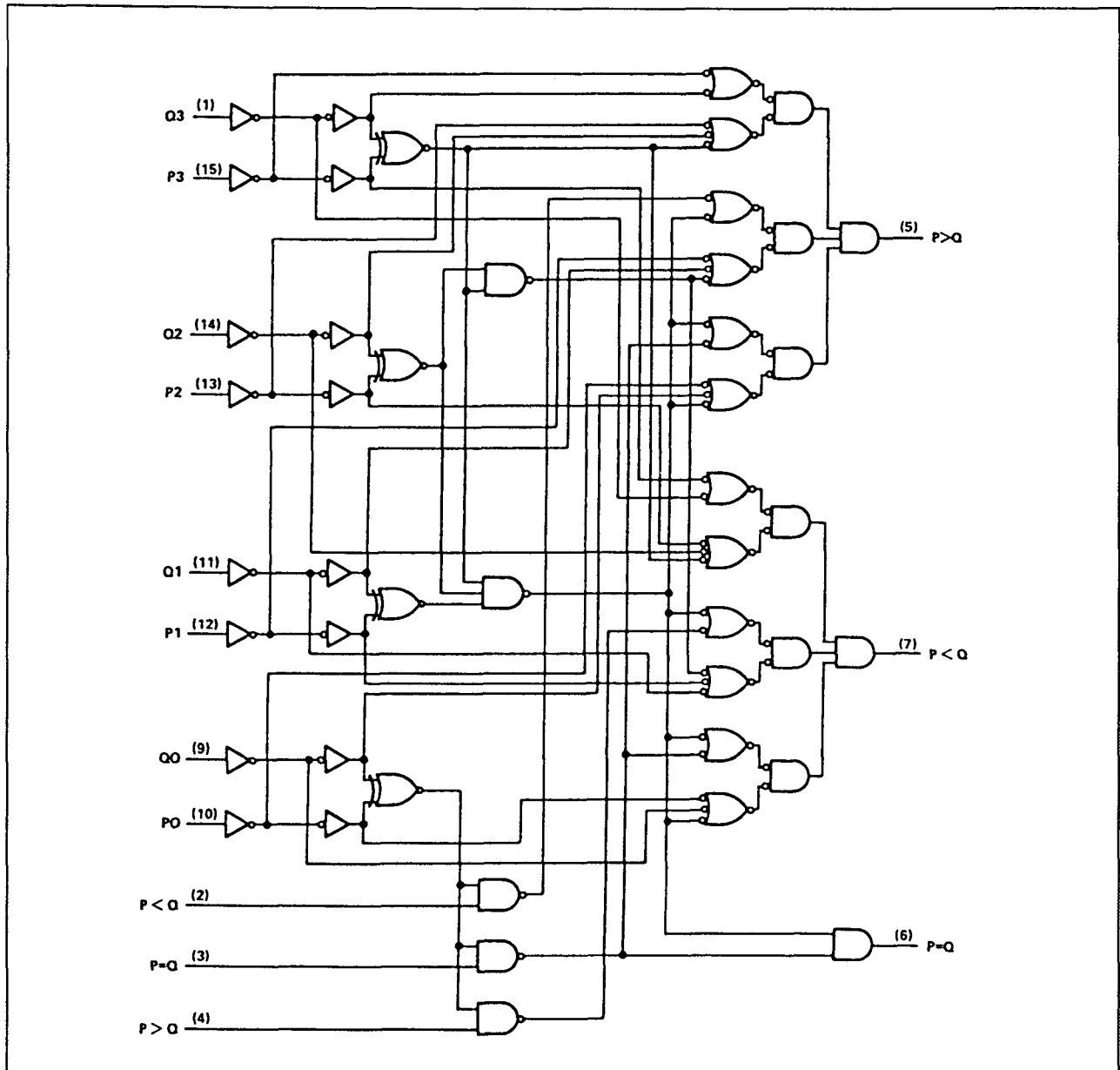
Zoals gezegd worden comparatoren gebruikt voor het vergelijken van binaire getallen. Bekende toepassingen daarvan zijn:

- het detecteren van een adres op een interfacekaart in een computer;
- het detecteren van een codewoord;
- het instellen van een teller.

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
P3, Q3	P2, Q2	P1, Q1	P0, Q0	P > Q	P < Q	P = Q	P > Q	P < Q	P = Q
P3 > Q3	X	X	X	X	X	X	H	L	L
P3 < Q3	X	X	X	X	X	X	L	H	L
P3 = Q3	P2 > Q2	X	X	X	X	X	H	L	L
P3 = Q3	P2 < Q2	X	X	X	X	X	L	H	L
P3 = Q3	P2 = Q2	P1 > Q1	X	X	X	X	H	L	L
P3 = Q3	P2 = Q2	P1 < Q1	X	X	X	X	L	H	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 > Q0	X	X	X	H	L	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 < Q0	X	X	X	L	H	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	H	L	L	H	L	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	L	H	L	L	H	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	X	X	H	L	L	H
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	H	H	L	L	L	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	L	L	L	H	H	L

Tabel 6/3.1-1: Waarheidstabel van de comparator 7485.

3.1 Achtergrond-informatie

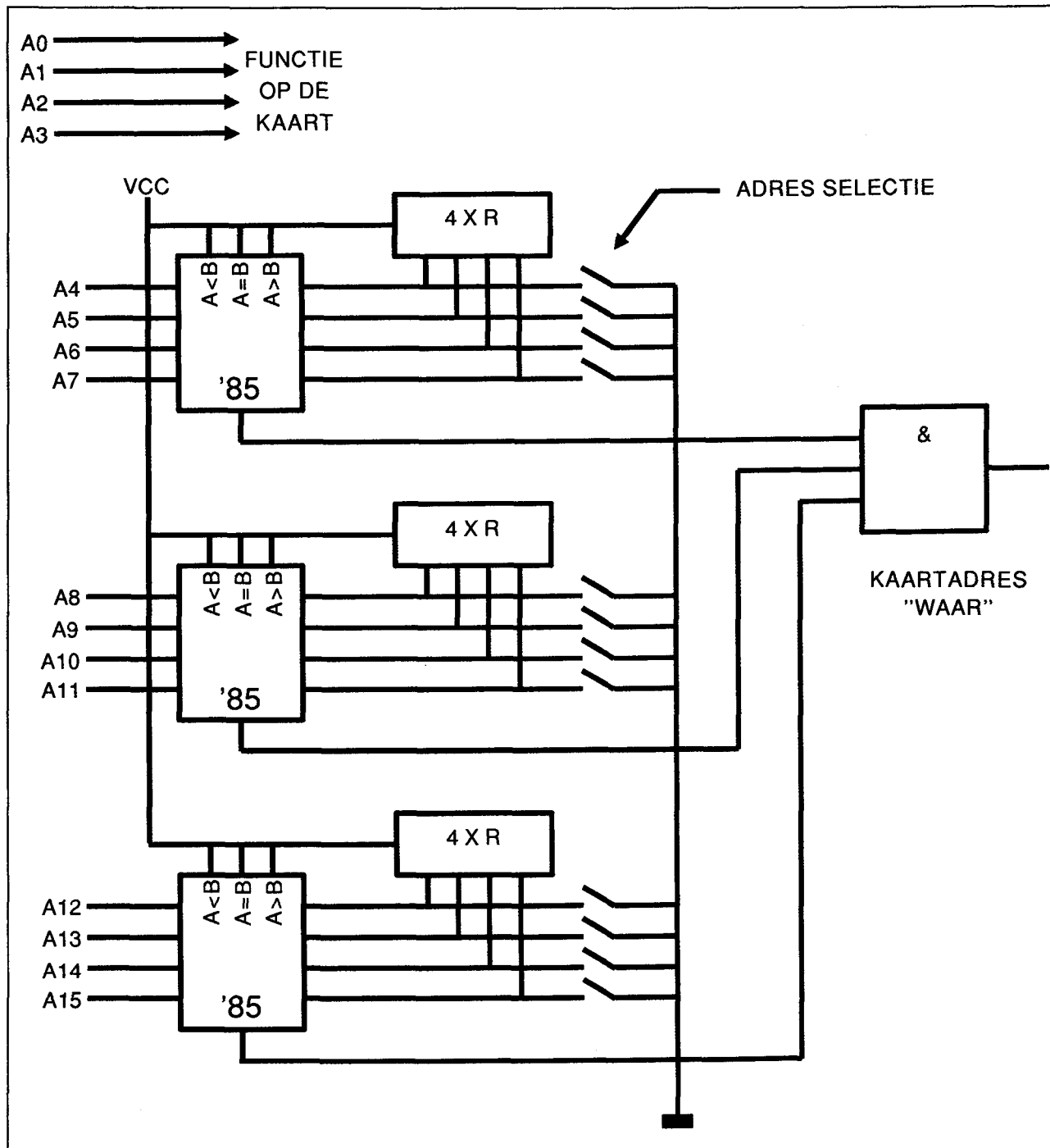


Figuur 6/3.1-4: Functioneel blokschema (positieve logika) van de 4-bit magnitude comparator 7485.

Wanneer de mogelijkheden van een computer worden vergroot door gebruik te maken van interfacekaarten (bijvoorbeeld een PIA-kaart, een floppy-disk controller of een geheugenkaart) moeten deze een eigen uniek adres hebben, zodat de betreffende kaart door het computerprogramma geactiveerd kan worden. Het instellen van zo'n adres gebeurt meestal met behulp van kleine schakelaars. In figuur 6/3.1-5 is te zien hoe van

een 16-bit adres 12 bits door de gebruiker kunnen worden ingesteld. De laagste 4 adresbits worden in dit geval door registers in de IC's op de kaart gebruikt (werking van de kaart), terwijl de overige 12 bits het kaart-adres vormen. Voor de duidelijkheid worden de te vergelijken woorden aan beide zijden van de comparatoren ingevoerd, links de adresbits van de bus en rechts de programmaschakelaars.

3.1 Achtergrond-informatie



Figuur 6/3.1-5: Programmering van 12 van de 16 adresbits op een interfacekaart.

In plaats van deze adresdetector kan ook een ander type zoals de 74ALS677 worden gebruikt (figuur 6/3.1-6). Hierbij worden de adreslijnen van de bus op de ingangen A1 tot en met A16 aangesloten. Met de P0- tot

en met P4-ingangen wordt (meestal door een "harde bedrading") het aantal te detecteren "nullen" ingesteld dat door de laagste adresingangen moet worden gedetecteerd. Wordt bijvoorbeeld de combinatie 0111 (de-

3.1 Achtergrond-informatie

cimaal 7) op de P-ingangen gezet, dan zijn de laagste 7 adresbits (A1 tot en met A7) ingericht om logische nullen te ontvangen. De resterende adresbits zijn dan ingesteld om "enen" te detecteren (zie ook de waarheidstabel 6/3.1-2). Het spreekt vanzelf dat het exacte adres wordt bepaald door de manier van aansluiten van de adreslijnen.

Fuse-programmable comparator

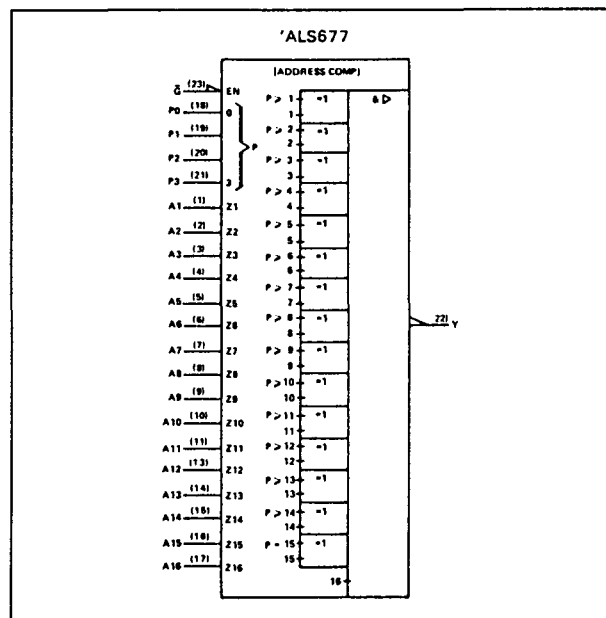
Een ander type comparator is de zogenaamde "fuse-programmable" identiteits comparator, zoals de 74ALS526 (zie figuur 6/3.1-7). Hierbij wordt een 16-bit woord (bijvoorbeeld een adres) vergeleken met een van te voren hierin geprogrammeerd datawoord. Het programmeren berust op het doorbranden van "zekeringen".

De bits worden geprogrammeerd (zekeringen doorgebrand) door een spanning van 12 V op zowel de betreffende P-ingang als de \overline{G} -ingang te zetten.

Hiervoor is geen speciale programmeer-apparatuur nodig.

Let op dat slechts één bit tegelijk mag worden geprogrammeerd en dat het programmeren

definitief is. De geprogrammeerde bits kunnen HOGE niveaus detecteren (logisch 1), de niet-geprogrammeerde herkennen LAGE niveaus (0).

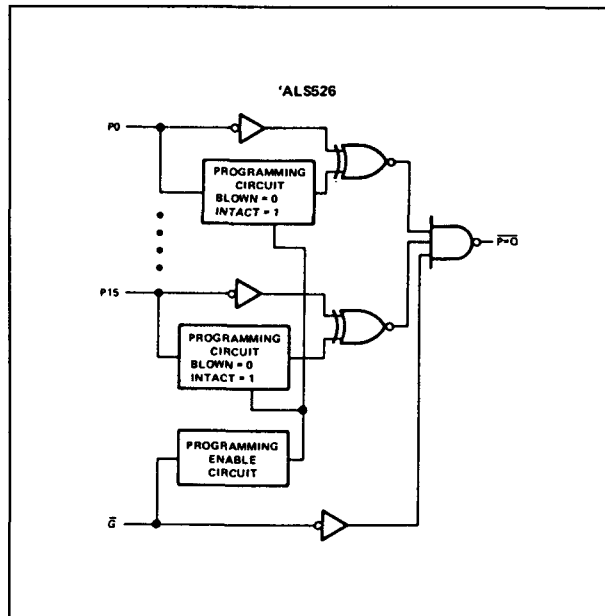


Figuur 6/3.1-6: Logisch symbool van de 16-bit adrescomparator 74ALS677.

[illegible]

Tabel 6/3.1-2: Waarheidstabel van de 74ALS677 adrescomparator.

3.1 Achtergrond-informatie



Figuur 6/3.1-7: Functioneel blokschema van de 16-bit fuse-programmable identiteitscomparator 74ALS526.

Op dezelfde wijze dat een adres kan worden gedetecteerd is het natuurlijk ook mogelijk om een willekeurig ander woord voor herkenning in te stellen. Zo kan bijvoorbeeld een elektronisch slot met een cijfercode worden geprogrammeerd.

Een andere toepassing is het tellen tot een bepaalde, in te stellen waarde. Het meest gebruikelijk is het laden van een teller met deze waarde, waarna wordt terug geteld tot nul. Wanneer dit echter niet mogelijk is, moet worden opgeteld totdat de geprogrammeerde waarde wordt bereikt.

Het schema van figuur 6/3.1-5 voldoet hieraan wanneer de A-ingangen op de uitgangen van de betreffende teller worden aangesloten.

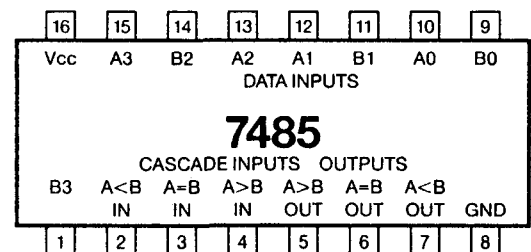
6/3.2

Magnitude comparatoren
74xx-serie TTL en HC

7485

4 bits magnitude comparator

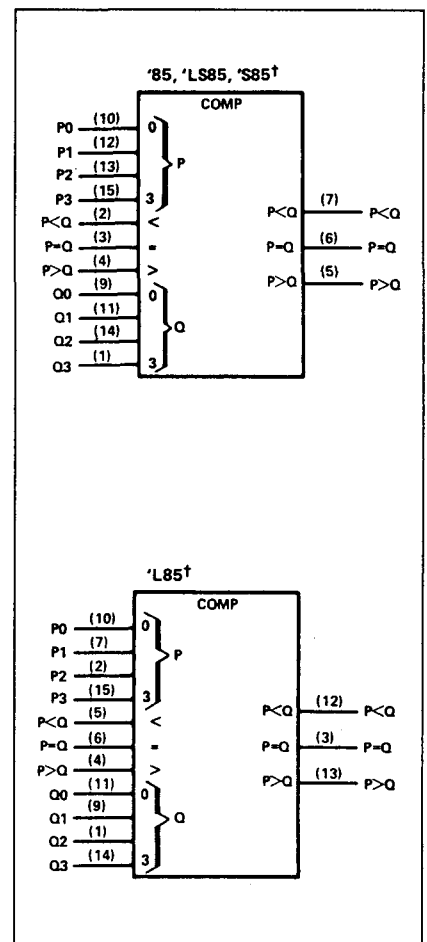
Figuur 3/3-85.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I_{cc}	H L	55 7.0	40 7.0	73 7.0	10.4 7.0			0.05 ³⁾	80 ³⁾	mA
I_{os}		-18 -55	-3 -15	-60 -150	-40 -100	-20 -100				mA
$T_{plh}^{2)}$		17	70	5 16	10.5	24		250	26	ns
$T_{phl}^{1)}$		20	55	5 16	11	20		250	26	ns
$T_{phl}^{2)}$		23	70	5 16	12	27		250	22	ns
$T_{phl}^{2)}$		20	55	5 13	11	23		250	22	ns

¹⁾ A, B to A>B, A<B (3 gate levels) ²⁾ A, B to A = B ³⁾ μA

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A3, B3	A2, B2	A1, B1	A0, B0	A>B	A<B	A=B	A>B	A<B	A=B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 > B3	A2 < B2	X	X	X	X	X	L	H	L
A3 > B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 > B3	A2 < B2	A1 < B1	X	X	X	X	L	H	L
A3 > B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 > B3	A2 < B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 > B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 < B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 < B3	A2 < B2	A1 = B1	A0 = B0	L	L	H	L	L	H
A3 < B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 < B3	A2 < B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 < B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

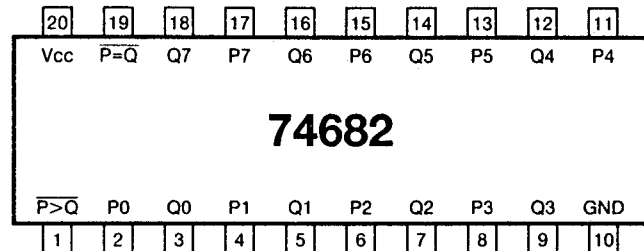


3.2 Magnitude comparatoren 74xx-serie TTL en HC

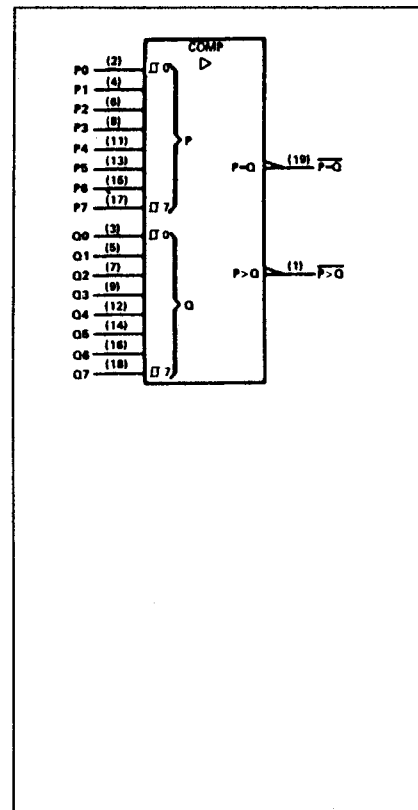
74682

8 bits magnitude comparator
20 k Ω pull-up op ingangen

Figuur 3/3-682.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}	H L				42				80 ³⁾	mA
I _{os}					-20 -100					mA
T _{plh} ¹⁾					13				26	ns
T _{phl} ¹⁾					15				26	ns
T _{phl} ²⁾					14				26	ns
T _{phl} ²⁾					15				26	ns
T _{phl} ²⁾					20				26	ns
T _{phl} ²⁾					15				26	ns
T _{phl} ²⁾					21				26	ns
T _{phl} ²⁾					19				26	ns



1) P to $\overline{P=Q}$ 2) Q to $\overline{P=Q}$ 3) P to $\overline{P>Q}$ 4) Q to $\overline{P<Q}$ 5) μA

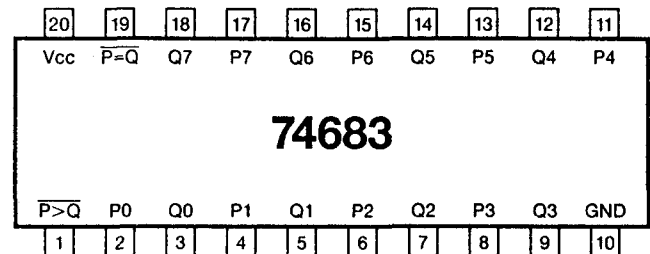
INPUTS			OUTPUTS	
DATA	ENABLES		$\overline{P=Q}$	$\overline{P>Q}$
P, Q	$\overline{Q_0}, \overline{Q_1}$	$\overline{Q_2}$		
P = Q	L	L	L	H
P > Q	L	L	H	L
P < Q	L	L	H	H
X	H	H	H	H

3.2 Magnitude comparatoren 74xx-serie TTL en HC

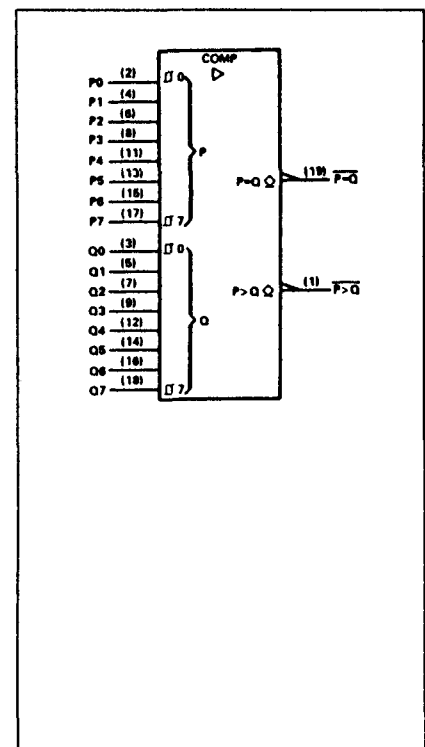
74683

8 bits magnitude comparator,
20 k Ω pull-up op ingangen,
open collector uitgang

Figuur 3/3-683.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I_{cc}	H L				42					mA
$T_{plh}^{(1)}$					30					ns
$T_{phl}^{(1)}$					20					ns
$T_{phl}^{(2)}$					24					ns
$T_{phl}^{(2)}$					23					ns
$T_{phl}^{(3)}$					31					ns
$T_{phl}^{(3)}$					17					ns
$T_{phl}^{(4)}$					30					ns
$T_{phl}^{(4)}$					21					ns



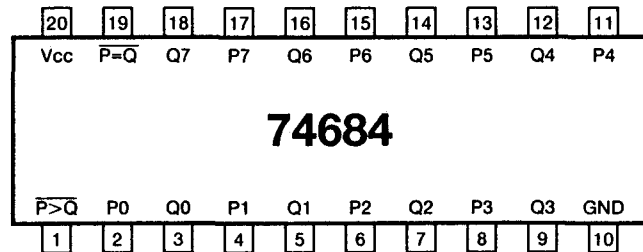
1) P to $\overline{P=Q}$ 2) Q to $\overline{P=Q}$ 3) P to $\overline{P>Q}$ 4) Q to $\overline{P>Q}$

INPUTS			OUTPUTS	
DATA	ENABLES		$\overline{P=Q}$	$\overline{P>Q}$
P, Q	$\overline{G_1}, G_1$	$\overline{G_2}, G_2$		
$P=Q$	L	L	L	H
$P>Q$	L	L	H	L
$P<Q$	L	L	H	H
X	H	H	H	H

3.2 Magnitude comparatoren 74xx-serie TTL en HC

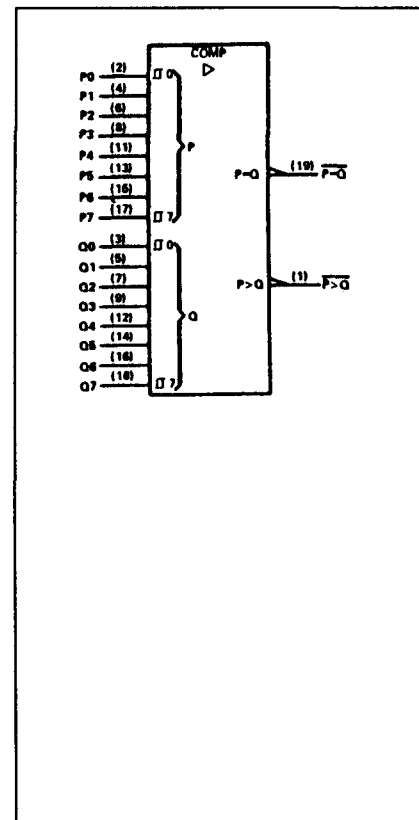
74684

8 bits magnitude comparator



Figuur 3/3-684.

LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I_{cc}	H L				40				80 ³⁾	mA
I_{os}					-20 -100					mA
$T_{plh}^{1)}$					15				26	ns
$T_{phl}^{1)}$					17				26	ns
$T_{phl}^{2)}$					16				26	ns
$T_{phl}^{2)}$					15				26	ns
$T_{phl}^{3)}$					22				26	ns
$T_{phl}^{3)}$					17				26	ns
$T_{phl}^{4)}$					24				26	ns
$T_{phl}^{4)}$					20				26	ns



1) P to $\overline{P=Q}$ 2) Q to $\overline{P=Q}$ 3) P to $\overline{P>Q}$ 4) Q to $\overline{P>Q}$ 5) μA

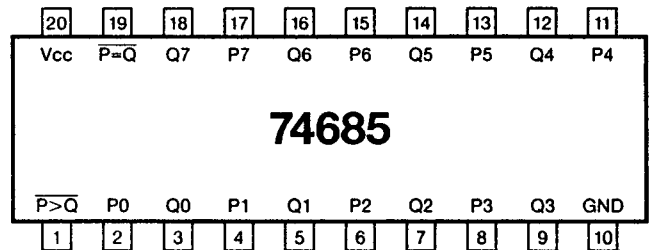
INPUTS			OUTPUTS	
DATA	ENABLES		$\overline{P=Q}$	$\overline{P>Q}$
P, Q	$\overline{G_1}, \overline{G_2}$	$\overline{G_3}$		
$P < Q$	L	L	L	H
$P > Q$	L	L	H	L
$P < Q$	L	L	H	H
X	H	H	H	H

3.2 Magnitude comparatoren 74xx-serie TTL en HC

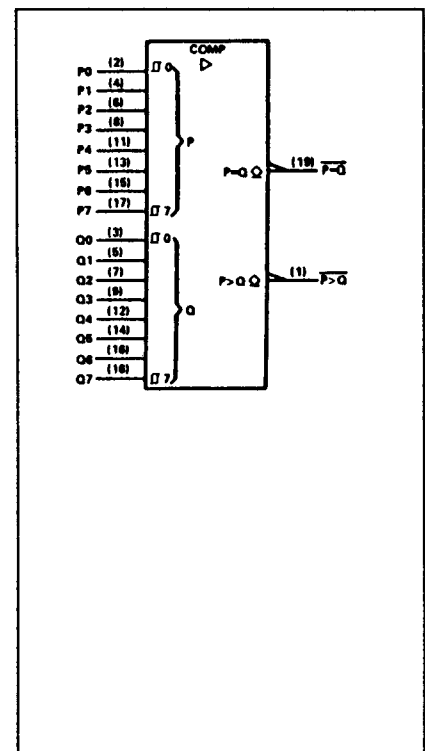
74685

8 bits magnitude comparator,
open collector uitgang

Figuur 3/3-685.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I_{cc}	H L				40					mA
$T_{plh}^{(1)}$					30					ns
$T_{phl}^{(1)}$					19					ns
$T_{phl}^{(2)}$					24					ns
$T_{phl}^{(2)}$					23					ns
$T_{phl}^{(3)}$					32					ns
$T_{phl}^{(3)}$					16					ns
$T_{phl}^{(4)}$					30					ns
$T_{phl}^{(4)}$					20					ns



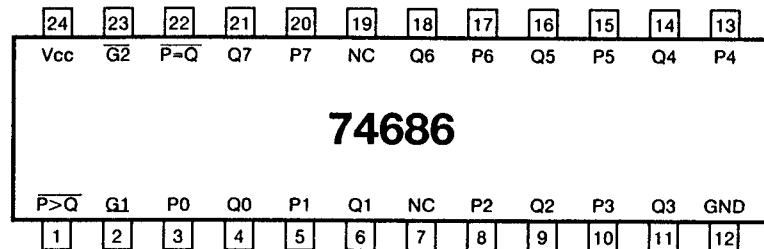
INPUTS			OUTPUTS	
DATA	ENABLES		$P=Q$	$P>Q$
P, Q	\bar{G}_1, \bar{G}_2	G_2		
$P=Q$	L	L	L	H
$P>Q$	L	L	H	L
$P<Q$	L	L	H	H
X	H	H	H	H

3.2 Magnitude comparatoren 74xx-serie TTL en HC

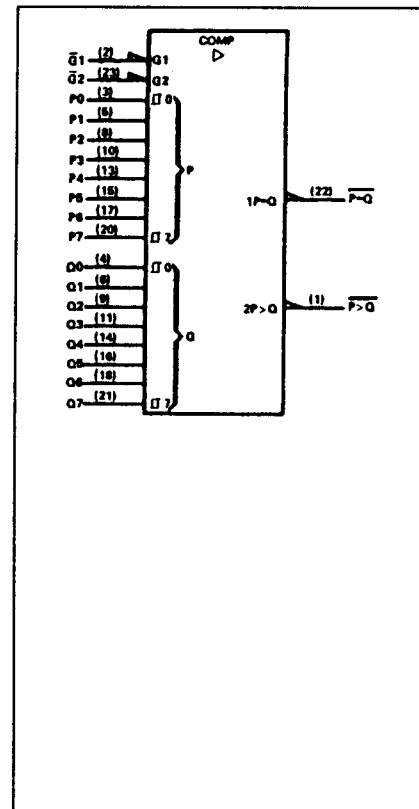
74686

8 bits magnitude comparator

Figuur 3/3-686.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABLE PARAMETERS										Eenheid
I_{cc}	H L				44				80 ⁵⁾	mA
I_{os}					-20 -100					mA
$T_{plh}^{1)}$					13				16	ns
$T_{phl}^{1)}$					20				16	ns
$T_{phl}^{2)}$					13				16	ns
$T_{phl}^{2)}$					21				16	ns
$T_{phl}^{3)}$					19				16	ns
$T_{phl}^{3)}$					15				16	ns
$T_{phl}^{4)}$					18				16	ns
$T_{phl}^{4)}$					19				16	ns



1) P to $\overline{P=Q}$ 2) Q to $\overline{P=Q}$ 3) P to $\overline{P>Q}$ 4) Q to $\overline{P>Q}$ 5) μA

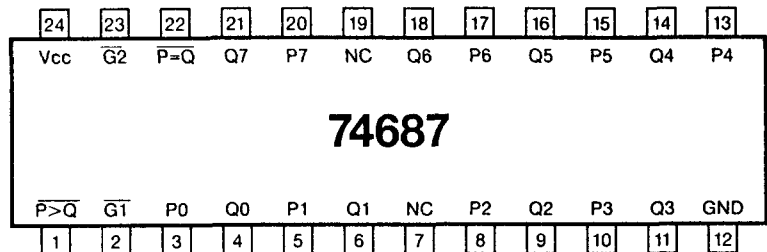
INPUTS			OUTPUTS	
DATA	ENABLES		$\overline{P=Q}$	$\overline{P>Q}$
P, Q	G1	G2		
P = Q	L	L	L	H
P > Q	L	L	H	L
P < Q	L	L	H	H
X	H	H	H	H

3.2 Magnitude comparatoren 74xx-serie TTL en HC

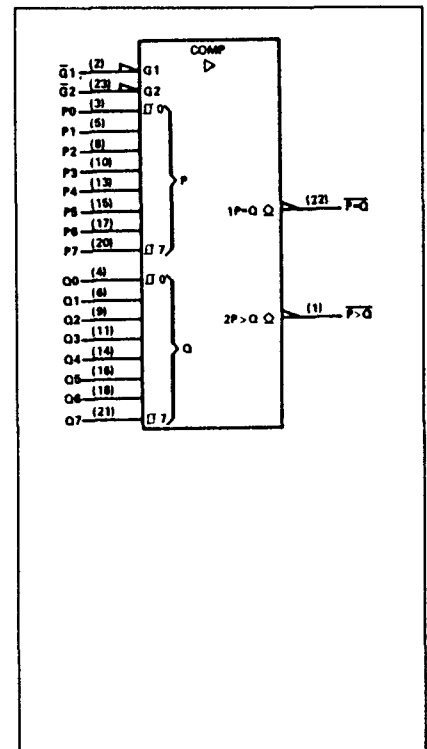
74687

8 bits magnitude
comparator, open
collector uitgang

Figuur 3/3-687.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}	H L				44					mA
T _{plh} ⁽¹⁾					24					ns
T _{phl} ⁽¹⁾					20					ns
T _{phl} ⁽²⁾					24					ns
T _{phl} ⁽²⁾					20					ns
T _{phl} ⁽³⁾					24					ns
T _{phl} ⁽³⁾					16					ns
T _{phl} ⁽⁴⁾					24					ns
T _{phl} ⁽⁴⁾					16					ns



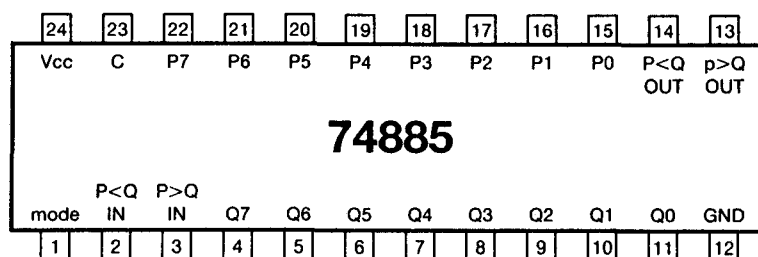
INPUTS			OUTPUTS	
DATA	ENABLES		P=Q	P>Q
P, Q	G, G1	G2		
P=Q	L	L	L	H
P>Q	L	L	H	L
P<Q	L	L	H	H
x	H	H	H	H

3.2 Magnitude comparatoren 74xx-serie TTL en HC

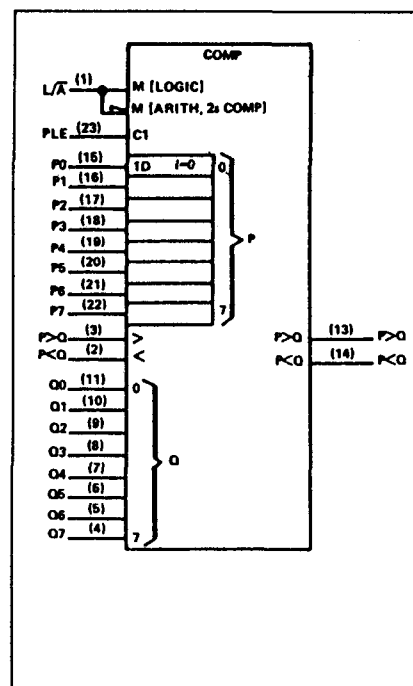
74885

8 bits magnitude comparator

Figuur 3/3-885.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABLE PARAMETERS										Eenheid
I _{cc}	H					130				mA
	L									
I _{os}						-20				mA
						-112				
T _{plh} ¹⁾						8.5				ns
T _{phl} ¹⁾						7.5				ns
T _{phl} ²⁾						5				ns
T _{phl} ²⁾						5.5				ns
T _{phl} ³⁾						13.5				ns
T _{phl} ³⁾						10				ns



¹⁾ L/A to p<Q, P>Q ²⁾ P<Q_{in} to P<Q, P>Q ³⁾ P,Q to P<Q, P>Q

COMPARISON	L/A	DATA INPUTS P0-P7, Q0-Q7	INPUT P>Q	INPUT P<Q	OUTPUTS P>Q P<Q
LOGICAL	H	P>Q	X	X	H L
LOGICAL	H	P<Q	X	X	L H
LOGICAL*	H	P=Q	H OR L	H OR L	H OR L H OR L
ARITHMETIC	L	P AG Q	X	X	H L
ARITHMETIC	L	Q AG P	X	X	L H
ARITHMETIC*	L	P=Q	H OR L	H OR L	H OR L H OR L

3.2 Magnitude comparatoren 74xx-serie TTL en HC

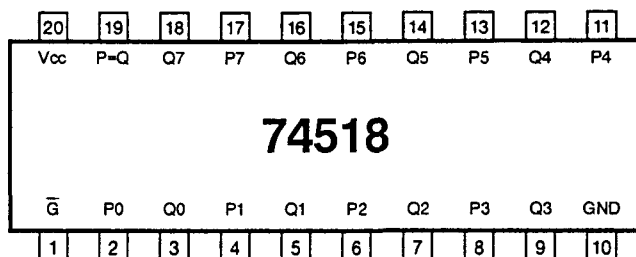
74518

8 bit identiteits

comparator

met 20 kΩ optrekweerstand aan Q-ingangen en open-collector uitgang

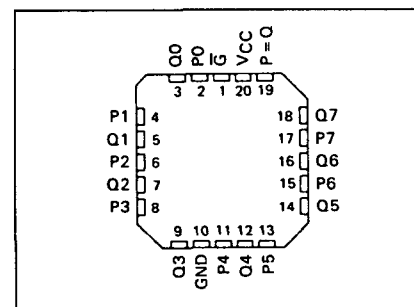
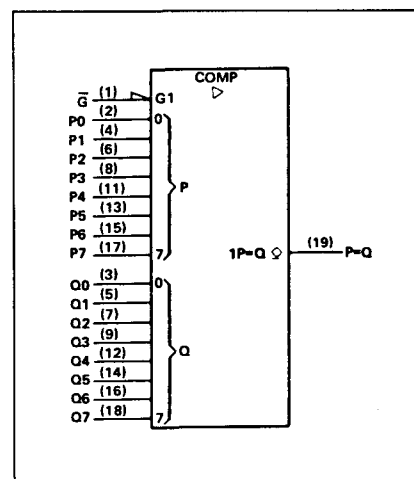
Figuur 6/3.2-518.



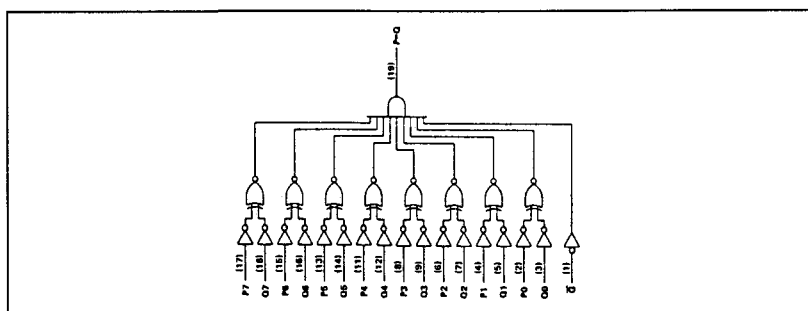
LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}			24				11			mA
I _{oi}			20				24			mA
T _{plh} ¹⁾			4				15			ns
T _{phl} ¹⁾			2				3			ns
T _{plh} ²⁾			4.5				15			ns
T _{phl} ²⁾			2				3			ns
			7.5				15			ns

1) P of Q → P=Q

2) \overline{G} → P=Q



Chip-carrier behuizing.

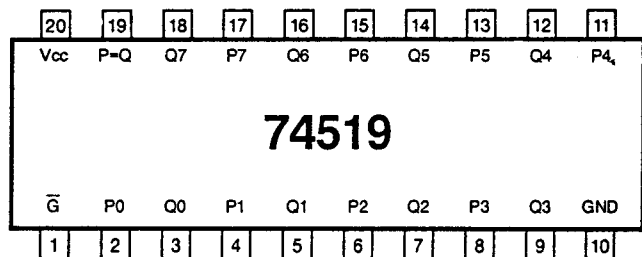


Functioneel blokschema (positieve logika).

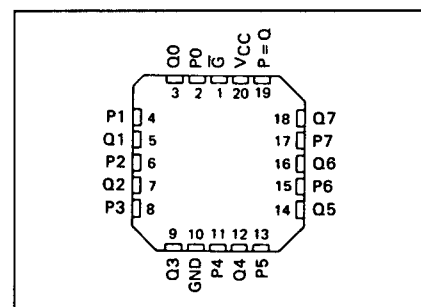
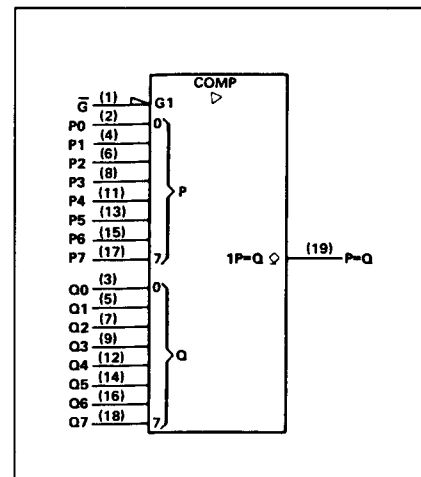
INPUTS		OUTPUT
DATA P, Q	ENABLE G	P = Q
P = Q	L	H
P > Q	L	L
P < Q	L	L
X	H	L

Waarheidstabel.

met open-collector uitgang

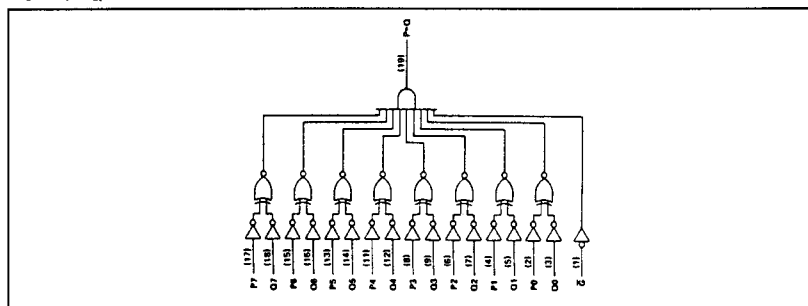


Figuur 6/3.2-519.

[illegible]

Chip-carrier behuizing.

1) P of Q \rightarrow P=Q

$$2) \overline{G} \rightarrow P=Q$$


Functioneel blokschema (positieve logika).

INPUTS		OUTPUT
DATA P, Q	ENABLE G	P = Q
P = Q	L	H
P > Q	L	L
P < Q	L	L
X	H	L

Waarheidstabel.

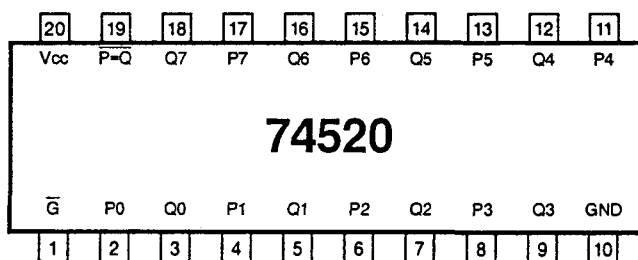
3.2 Magnitude comparatoren 74xx-serie TTL en HC

74520

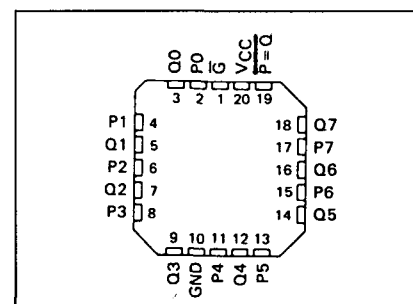
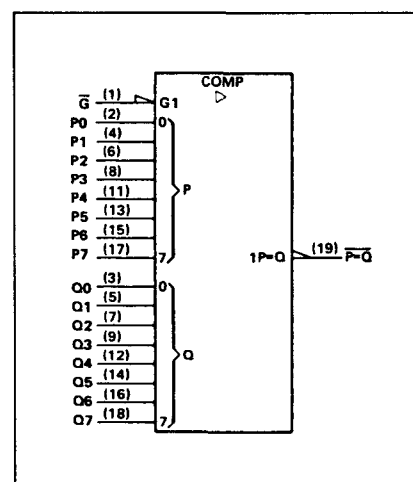
8 bits identiteits comparator

met 20 k Ω optrekweerstand aan Q-ingangen en geïnverteerde totempaal-uitgang

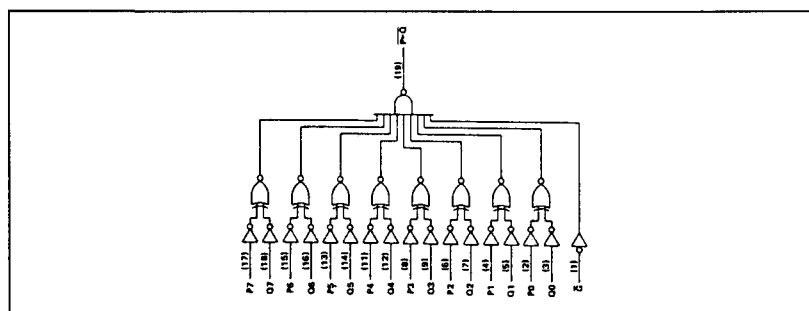
Fguur 6/3.2-520.

[illegible]

1) $P \text{ of } Q \rightarrow \overline{P=Q}$

$$2) \overline{G} \rightarrow \overline{P=Q}$$


Chip-carrier behuizing.



Functioneel blokschema (positieve logika).

INPUTS		OUTPUT
DATA P, Q	ENABLE G	$\overline{P=Q}$
P=Q	L	L
P>Q	L	H
P<Q	L	H
X	H	H

Waarheidstabel.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

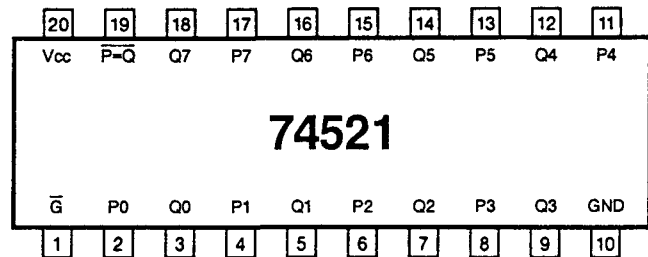
74521

8 bit identiteits

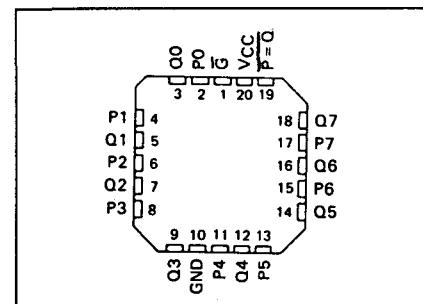
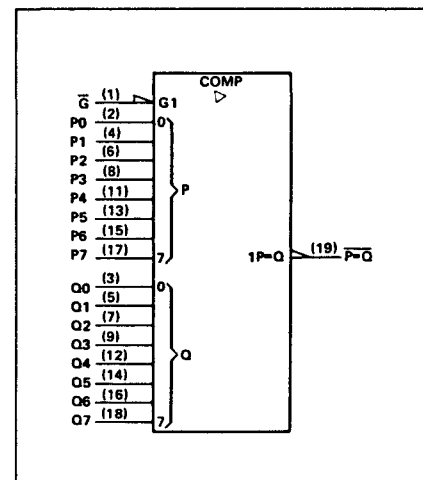
comparator

met geïnverteerde totempaal-uitgang

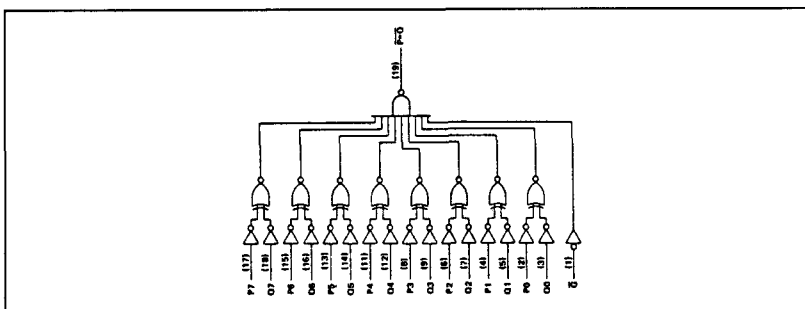
Figuur 6/3.2-521.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}			21				12		80 ³⁾	mA
I _{os}			-60				-30		4	mA
T _{plh} ¹⁾			2.7				3		21	ns
T _{phl} ¹⁾			11				12		21	ns
T _{plh} ²⁾			2.2				2		14	ns
T _{phl} ²⁾			7.5				12		14	ns
			10				22			

1) P of Q → $\overline{P-Q}$ 2) $\overline{G} \rightarrow \overline{P-Q}$ 3) μA 

Chip-carrier behuizing.



Functioneel blokschema (positieve logika).

INPUTS		OUTPUT:
DATA P, Q	ENABLE G	$\overline{P=Q}$
P=Q	L	L
P>Q	L	H
P<Q	L	H
X	H	H

Waarheidstabel.

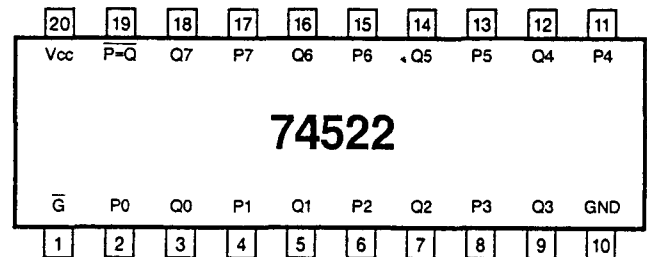
3.2 Magnitude comparatoren 74xx-serie TTL en HC

74522

8 bit identiteits
comparator

met 20 k Ω optrekweerstanden aan
Q-ingangen en open-collector
uitgang (geïnverteerd)

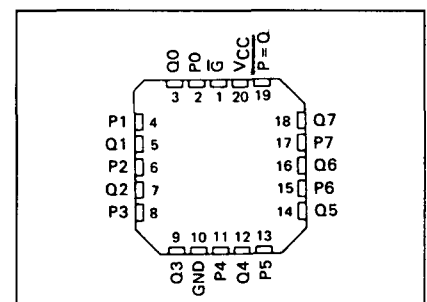
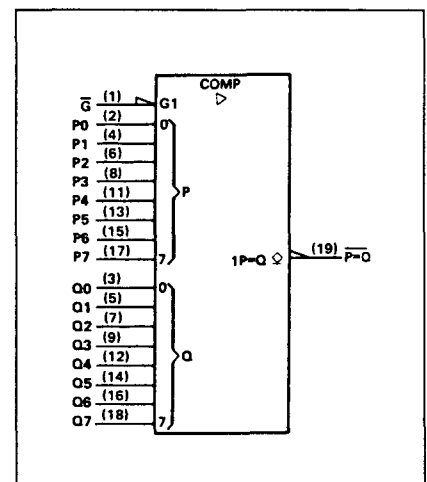
Figuur 6/3.2-522.



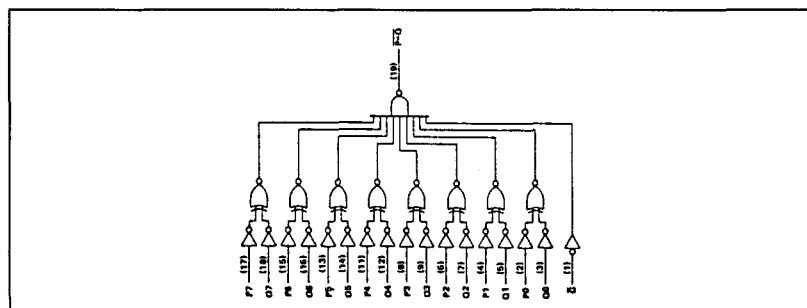
LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABLE PARAMETERS										Eenheid
I _{cc}							11			mA
I _{ol}							24			mA
T _{plh} ¹⁾							10			ns
T _{phl} ¹⁾							5			ns
T _{plh} ²⁾							8			ns
T _{phl} ²⁾							8			ns
							23			ns
							25			ns
							8			ns
							23			ns

¹⁾ P of Q \rightarrow $\overline{P=Q}$

²⁾ $\overline{G} \rightarrow \overline{P=Q}$



Chip-carrier behuizing.



Functioneel blokschema (positieve logica).

INPUTS		OUTPUT
DATA P, Q	ENABLE G	$\overline{P=Q}$
P=Q	L	L
P>Q	L	H
P<Q	L	H
X	H	H

Waarheidstabel.

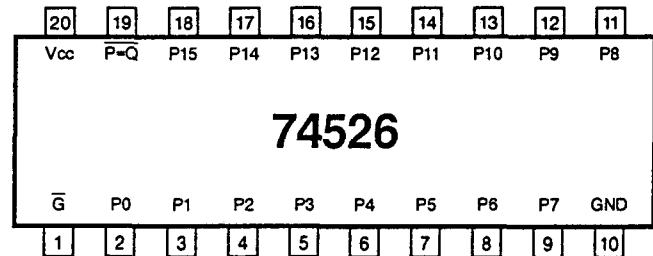
3.2 Magnitude comparatoren 74xx-serie TTL en HC

74526

16 bit fuse programmable
identiteits comparator

programmeren (HOOG herkennen) door 12 V op gewenste P-pen en \bar{G} te zetten)

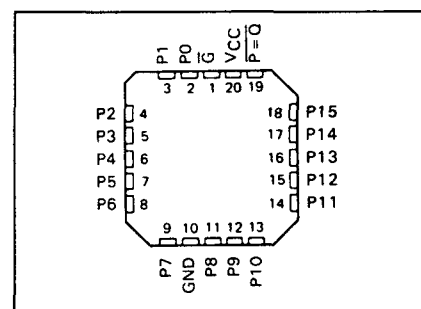
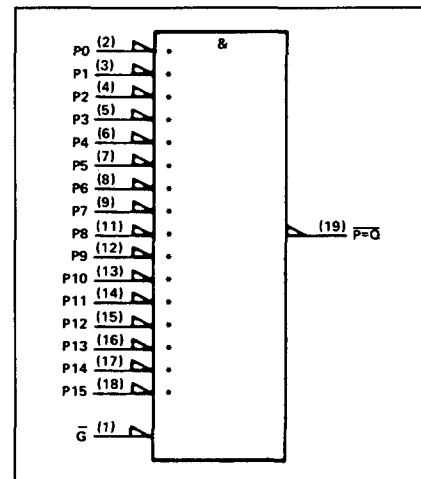
Figuur 6/3.2-526.



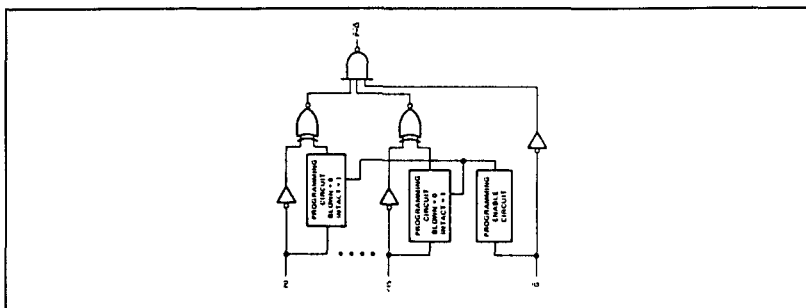
LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I_{cc}							16			mA
I_{os}							-30			mA
T_{plh}^1							3			ns
T_{phl}^1							15			ns
T_{plh}^2							2			ns
T_{phl}^2							12			ns
							2			ns
							15			ns
							12			ns

¹⁾ P of Q $\rightarrow \overline{P-Q}$

²⁾ $\bar{G} \rightarrow \overline{P-Q}$



Chip-carrier behuizing.

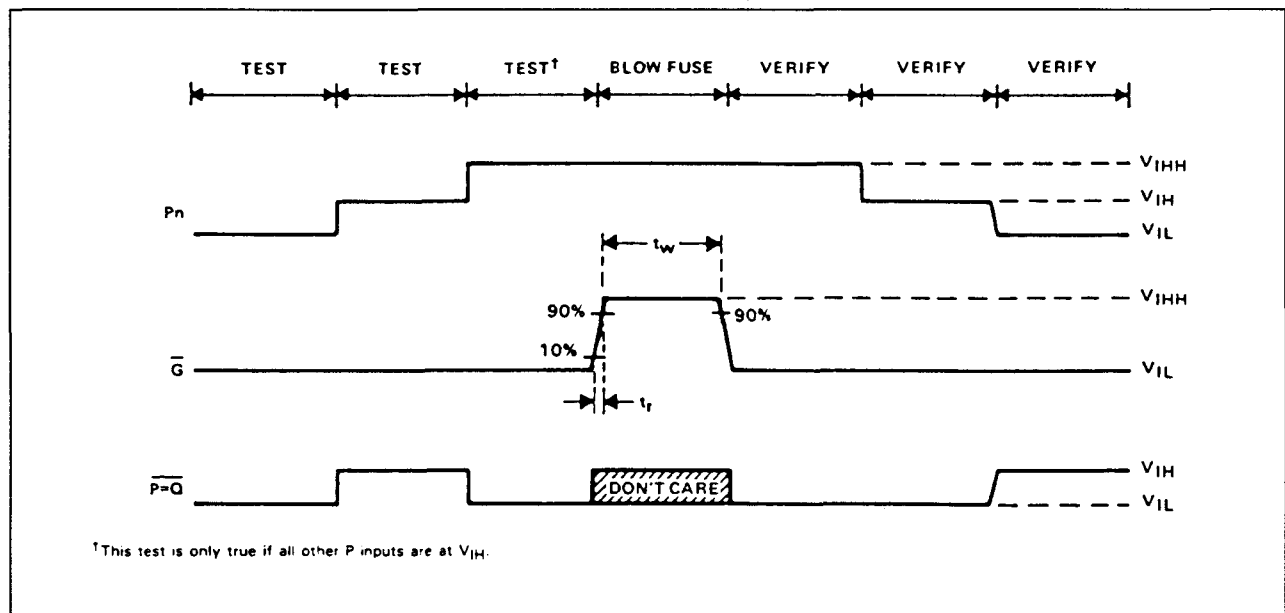


Functioneel blokschema (positieve logika).

3.2 Magnitude comparatoren 74xx-serie TTL en HC

PARAMETER		MIN	MAX	UNIT
V_{IH}	High-level input voltage	2	5.5	V
V_{IL}	Low-level input voltage		0.8	V
V_{IHH}	Program-pulse input voltage	11.5	12.5	V
V_{CC}	Supply voltage	6.5	7.5	V
I_{IHH}	Program-pulse input current	$P_n (\bar{G} \text{ low})$	10	mA
		\bar{G}	1.24	
I_{CCHH}	Supply current with V_{IHH} applied	ALS526	31	mA
t_w	Pulse duration, program	10	50	μs
t_r	Rise time, program voltage		10	μs

Programmeer parameters.



Programmeer timing.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

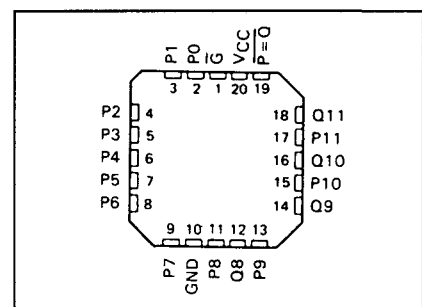
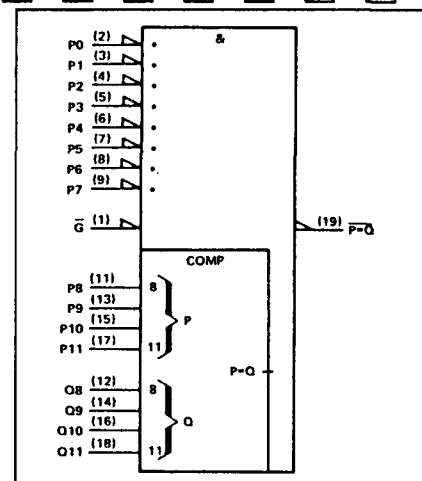
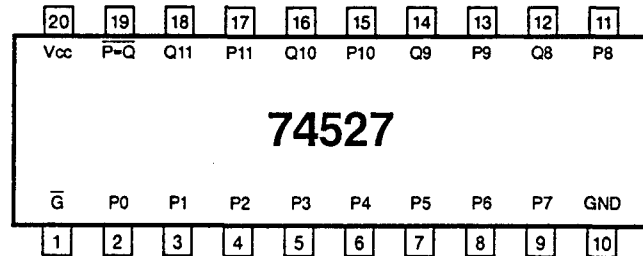
74527

8 bit fuse programmable
identiteits comparator en 4 bit
comparator

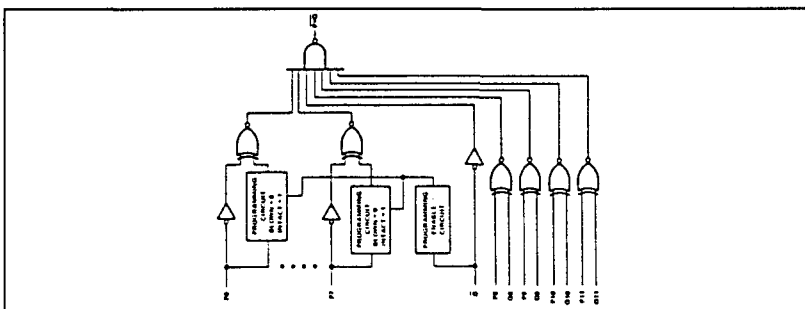
programmeren (H herkennen) door 12 V op
gewenste P-pen en \bar{G} te zetten

Figuur 6/3.2-527.

LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Een- heid
I _{cc}							15			mA
I _{os}							-30 -130			mA
T _{plh} ¹⁾							3 15			ns
T _{phl} ¹⁾							2 15			ns
T _{plh} ²⁾							12 2			ns
T _{phl} ²⁾							15 2 12			ns

1) P of Q → $\overline{P=Q}$ 2) $\bar{G} \rightarrow \overline{P=Q}$ 

Chip-carrier behuizing.

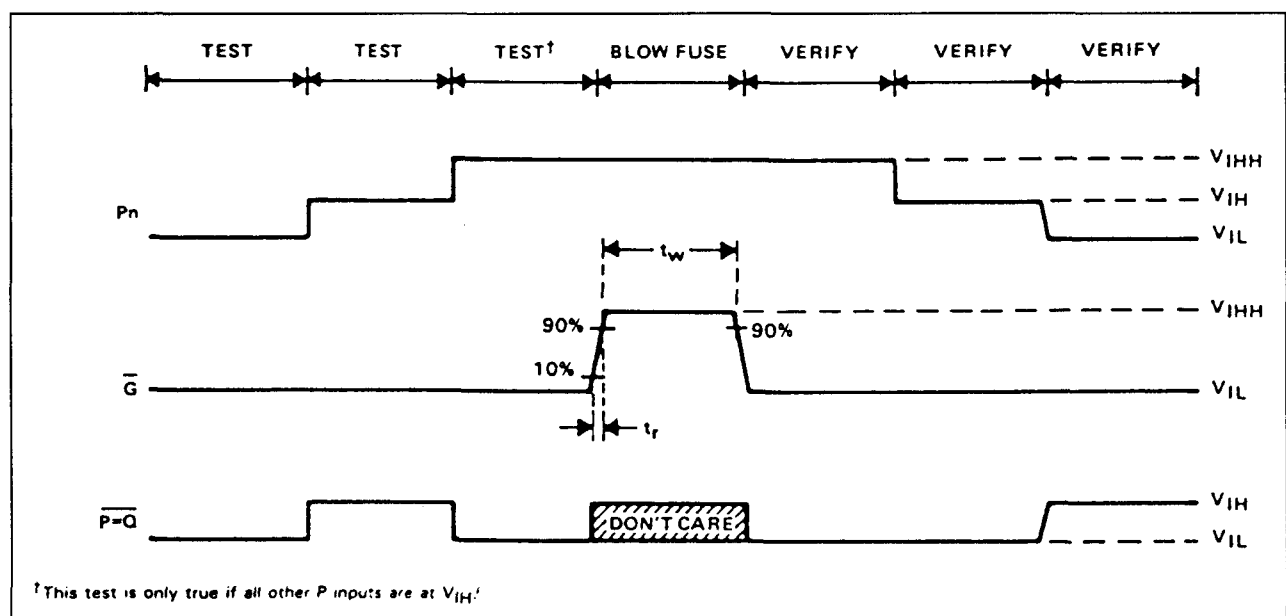


Functioneel blokschema (positieve logica).

3.2 Magnitude comparatoren 74xx-serie TTL en HC

PARAMETER		MIN	MAX	UNIT
V_{IH}	High-level input voltage	2	5.5	V
V_{IL}	Low-level input voltage		0.8	V
V_{IHH}	Program-pulse input voltage	11.5	12.5	V
V_{CC}	Supply voltage	6.5	7.5	V
I_{IHH}	Program-pulse input current	$P_n (\bar{G} \text{ low})$	10	mA
		\bar{G}	1.24	
I_{CCHH}	Supply current with V_{IHH} applied			mA
		ALS527	29	
t_w	Pulse duration, program	10	50	μs
t_r	Rise time, program voltage		10	μs

Programmeer parameters.



Programmeer timing.

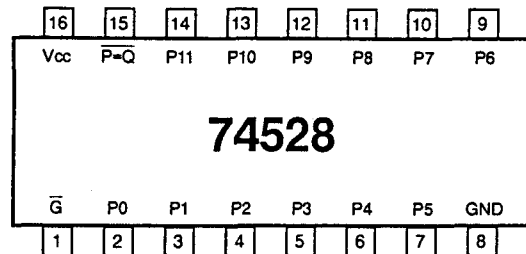
3.2 Magnitude comparatoren 74xx-serie TTL en HC

74528

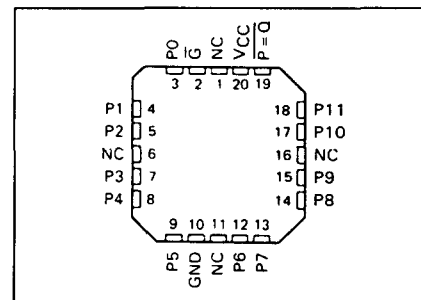
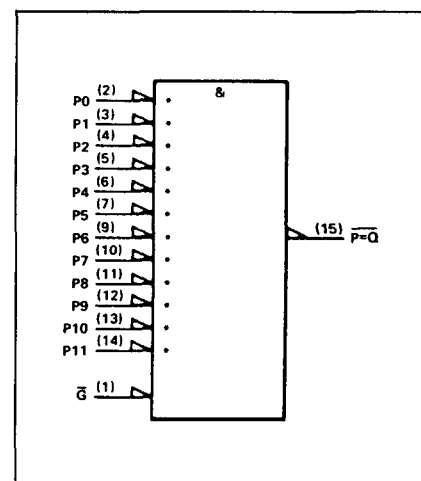
12 bit fuse programmable
identiteits comparator

programmeren (HOOG herkennen) door 12 V
op gewenste P-pen en \bar{G} te zetten

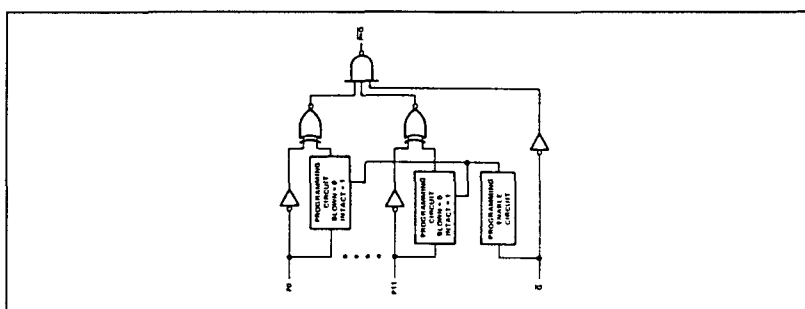
Figuur 6/3.2-528.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}							13			mA
I _{os}							-30			mA
T _{plh} ¹⁾							3			ns
T _{phl} ¹⁾							15			ns
T _{plh} ²⁾							2			ns
T _{phl} ²⁾							12			ns
							15			ns
							2			ns
							12			ns



Chip-carrier behuizing.

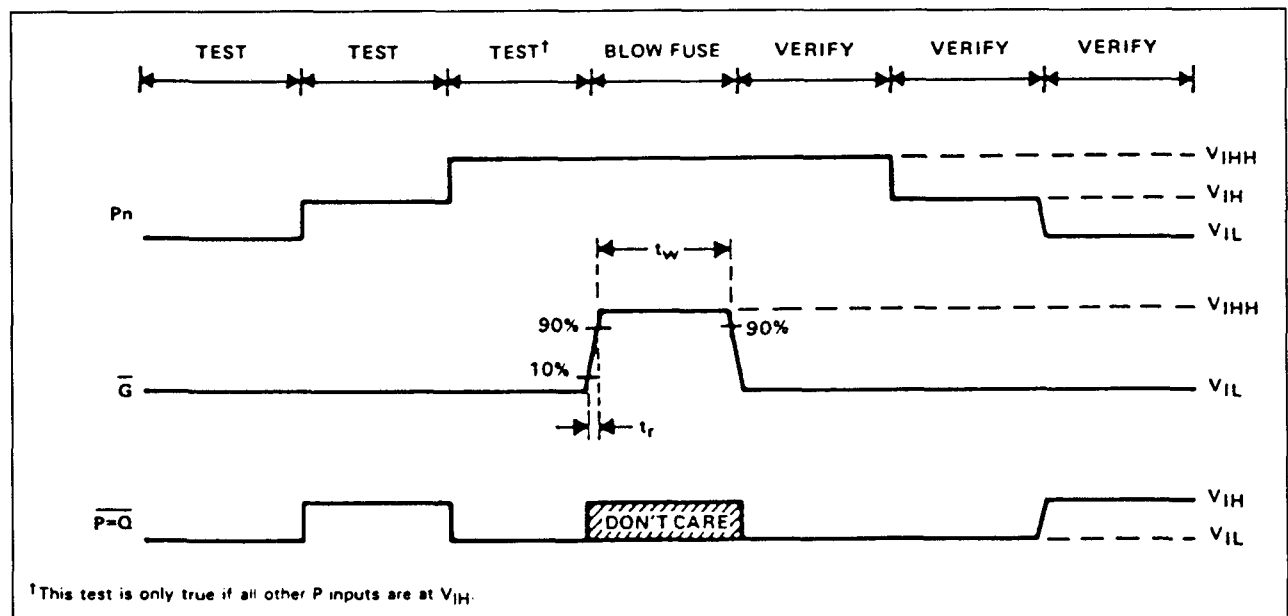
1) $P \text{ of } Q \rightarrow \overline{P=Q}$ 2) $\bar{G} \rightarrow P=Q$ 

Functioneel blokschema (positieve logica).

3.2 Magnitude comparatoren 74xx-serie TTL en HC

PARAMETER		MIN	MAX	UNIT
V_{IH}	High-level input voltage	2	5.5	V
V_{IL}	Low-level input voltage		0.8	V
V_{IHH}	Program-pulse input voltage	11.5	12.5	V
V_{CC}	Supply voltage	6.5	7.5	V
I_{IHH}	Program-pulse input current	$P_n (\bar{G} \text{ low})$	10	mA
		\bar{G}	1.24	
I_{CCHH}	Supply current with V_{IHH} applied			mA
		'ALS528	26	
t_w	Pulse duration, program	10	50	μs
t_r	Rise time, program voltage		10	μs

Programmeer parameters.



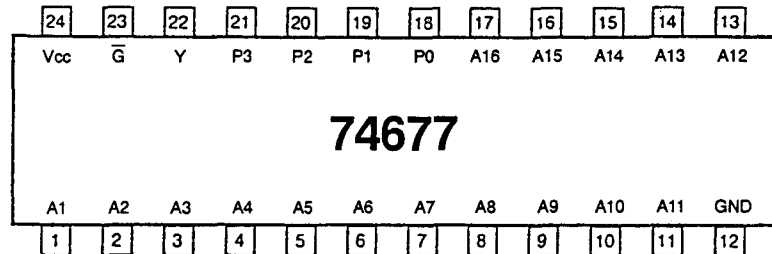
Programmeer timing.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

74677**16 bit adres comparator met enable**

(aantal te detecteren LAGE bits instelbaar met P0 t/m P3 voor laagste adresbits)

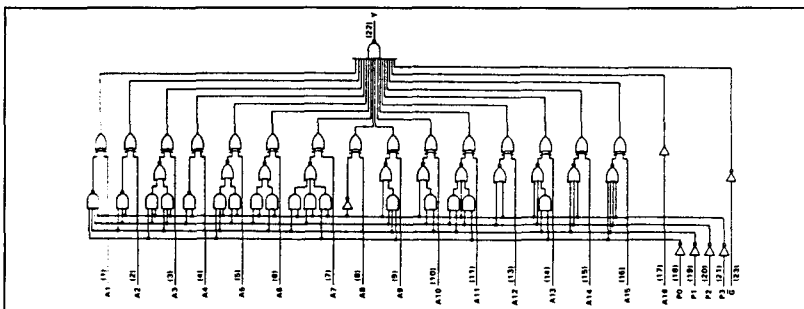
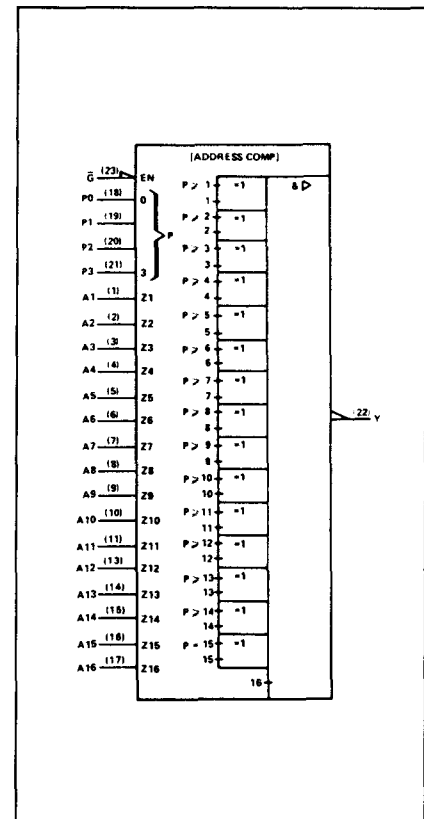
Figuur 6/3.2-677.



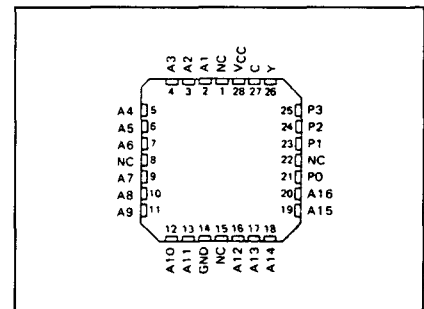
LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}							21		80 ⁴⁾	mA
I _{os}							-30		4	mA
T _{plh} ¹⁾							4		50	ns
T _{phl} ¹⁾							25		50	ns
T _{plh} ²⁾							8		18	ns
T _{phl} ²⁾							35		18	ns
T _{plh} ³⁾							5		14	ns
T _{phl} ³⁾							22		14	ns
							5			
							35			
							3			
							13			
							5			
							25			

1) P → y

2) A → y

3) $\bar{Q} \rightarrow y$ 4) μA 

Functioneel blokschema (positieve logika).



Chip-carrier behuizing.

'ALS677	INPUTS COMMON TO 'ALS677																OUTPUT				
G	P3	P2	P1	P0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	Y
L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	L	L	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	L	L	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	L	H	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	L	H	L	H	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	L
L	L	H	H	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	L
L	L	H	H	H	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	L
L	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	L
L	H	L	L	H	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	L
L	H	L	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	L
L	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	L
L	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	L
L	H	H	L	H	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	L
L	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	L
L	All other combinations																				H
H	'ALS677: Any combination																				H

Waarheidstabel.

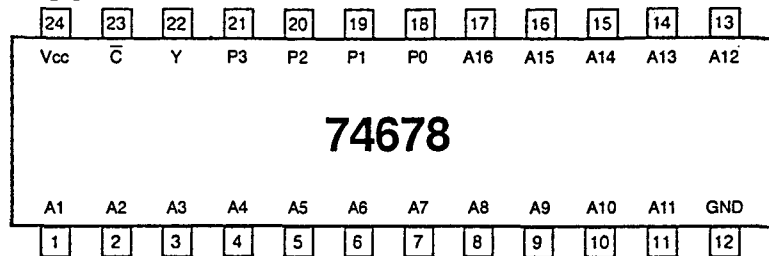
3.2 Magnitude comparatoren 74xx-serie TTL en HC

74678

16 bit adres comparator met latch

(aantal te detecteren LAGE bits instelbaar met P0 t/m P3 voor laagste adresbits)

Figuur 6/3.2-678.



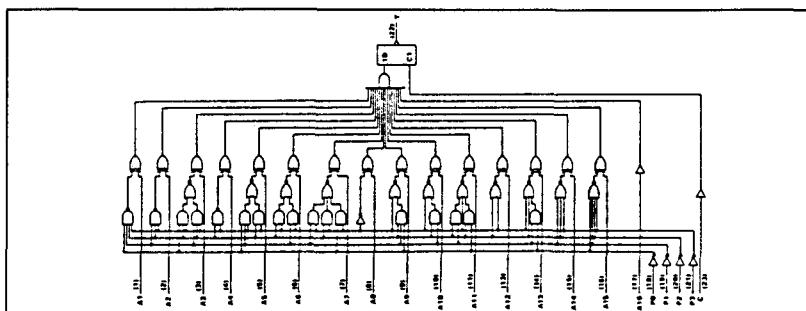
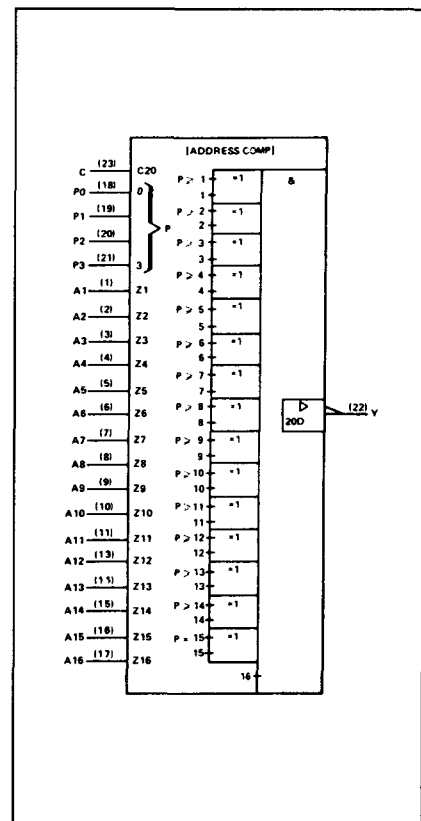
LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}							21		80 ⁴⁾	mA
I _{os}							-30		4	mA
T _{plh} ¹⁾							6		50	ns
T _{phl} ¹⁾							22		50	ns
T _{plh} ²⁾							10		23	ns
T _{phl} ²⁾							43		23	ns
T _{plh} ³⁾							5		19	ns
T _{phl} ³⁾							21		19	ns
							5			
							35			
							3			
							20			
							15			
							48			

1) P → y

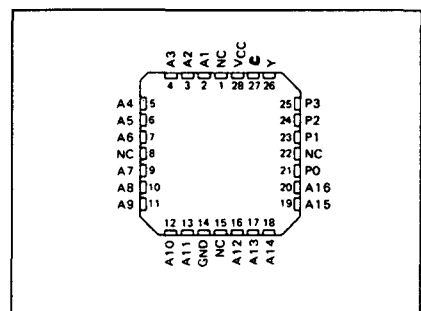
2) A → y

3) C → y

4) μA



Functioneel blokschema (positieve logika).



Chip-carrier behuizing.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

'ALS678 C	INPUTS COMMON TO 'ALS678																OUTPUT				
	P3	P2	P1	P0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	Y
H	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
H	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
H	L	L	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
H	L	L	H	H	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L
H	L	H	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L
H	L	H	L	H	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	H	H	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	L
H	L	H	H	H	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	L
H	L	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	L
H	L	H	L	H	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	L
H	L	H	L	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	L
H	L	H	L	H	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	L
H	L	H	L	H	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	L
H	L	H	L	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	L
H	L	H	L	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	L
L	'ALS678: Any combination																				Latched

Waarheidstabel.

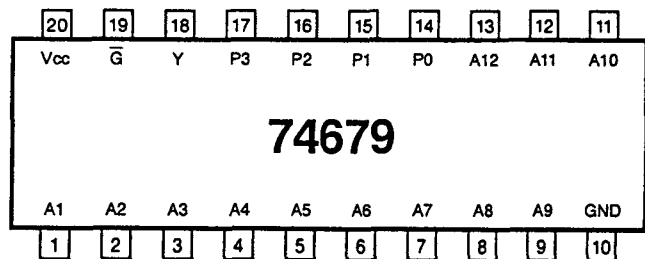
3.2 Magnitude comparatoren 74xx-serie TTL en HC

74679

12 bit adres comparator met enable

(aantal te detecteren LAGE bits instelbaar met P0 t/m P3 voor laagste adresbits)

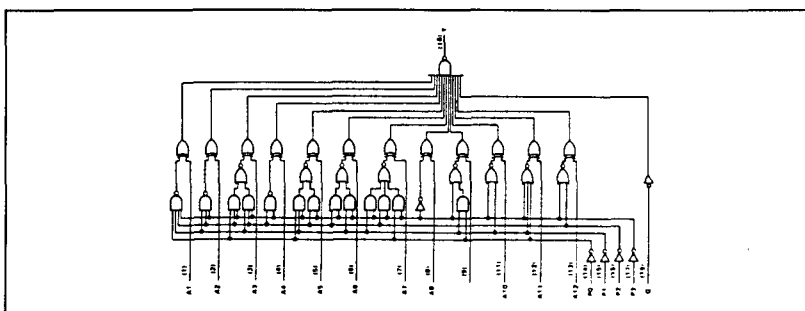
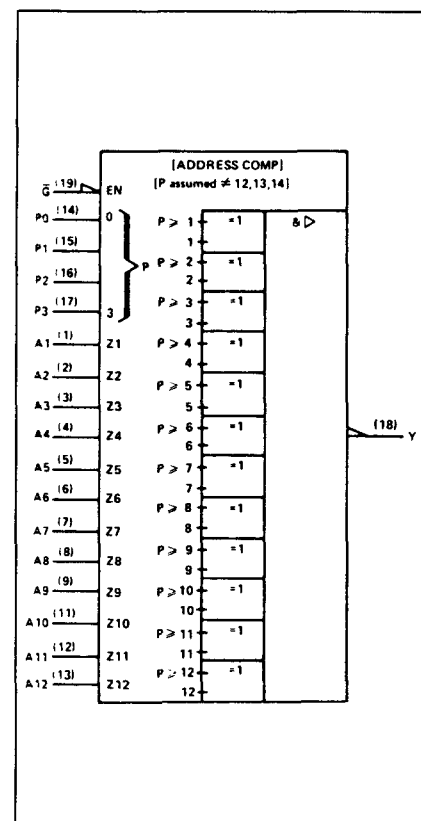
Figuur 6/3.2-679.



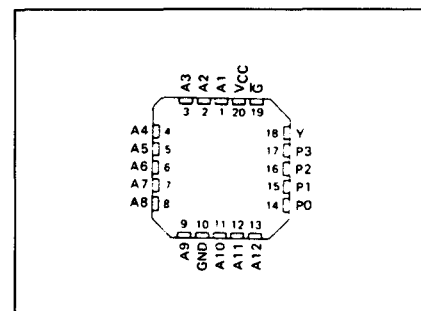
LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}							17		80 ⁴⁾	mA
I _{os}							-30		4	mA
T _{plh} ¹⁾							4		37	ns
T _{phl} ¹⁾							8		37	ns
T _{plh} ²⁾							35		21	ns
T _{phl} ²⁾							5		21	ns
T _{plh} ³⁾							22		15	ns
T _{phl} ³⁾							5		15	ns
							30			
							13			
							5			
							25			

1) P → y

2) A → y

3) $\bar{G} \rightarrow y$ 4) μA 

Functioneel blokschema (positieve logica).



Chip-carrier behuizing.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

'ALS679	INPUTS COMMON TO 'ALS679												OUTPUT				
G	P3	P2	P1	P0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Y
L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	L
L	L	L	H	L	L	L	H	H	H	H	H	H	H	H	H	H	L
L	L	L	H	H	L	L	L	H	H	H	H	H	H	H	H	H	L
L	L	H	L	L	L	L	L	L	H	H	H	H	H	H	H	H	L
L	L	H	L	H	L	L	L	L	L	H	H	H	H	H	H	H	L
L	L	H	H	L	L	L	L	L	L	L	H	H	H	H	H	H	L
L	L	H	H	H	L	L	L	L	L	L	L	H	H	H	H	H	L
L	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	L
L	H	L	L	H	L	L	L	L	L	L	L	L	L	H	H	H	L
L	H	L	H	L	L	L	L	L	L	L	L	L	L	L	H	H	L
L	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	H	L
L	H	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	L*
L	H	H	L	H	L	L	L	L	L	L	L	L	L	H	H	L	L*
L	H	H	H	L	L	L	L	L	L	L	L	L	L	L	H	L	L*
L	H	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L
L	All other combinations																H
H	'ALS679: Any combination																H

*The three shaded rows of the function table show combinations that would normally not be used in address comparator applications. The logic symbols above are not valid for these combinations in which P = 12, 13, and 14. If symbols valid for all combinations are required, starting with the fourth Exclusive-OR from the bottom, change P ≥ 9 to P = 9...11/13...15, P ≥ 10 to P = 10/11/14 15, and P ≥ 11 to P = 11/15.

Waarheidstabel.

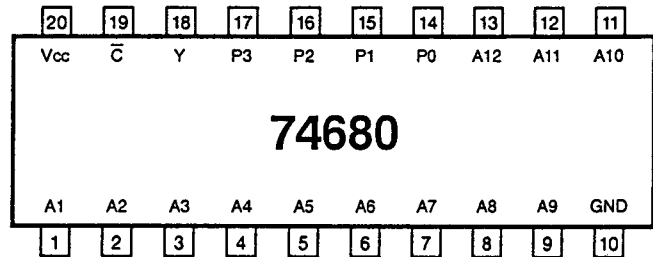
3.2 Magnitude comparatoren 74xx-serie TTL en HC

74680

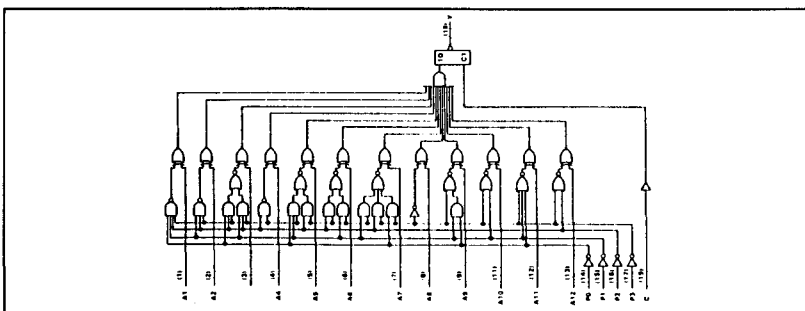
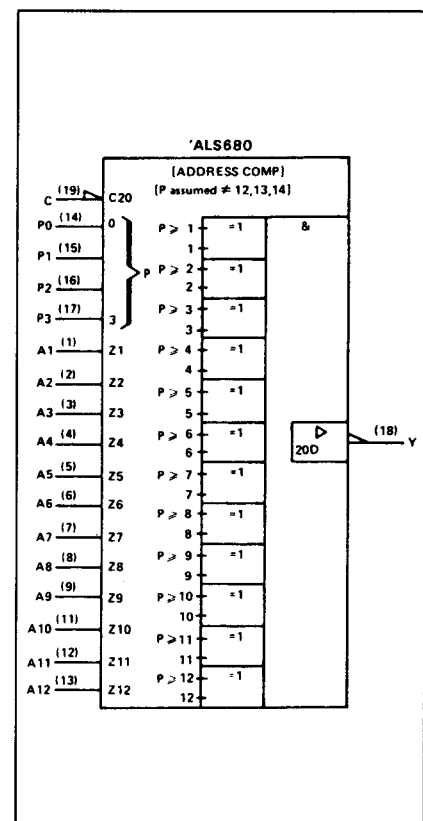
12 bit adrescomparator met latch

(aantal te detecteren LAGE bits instelbaar met P0 t/m P3 voor laagste adresbits)

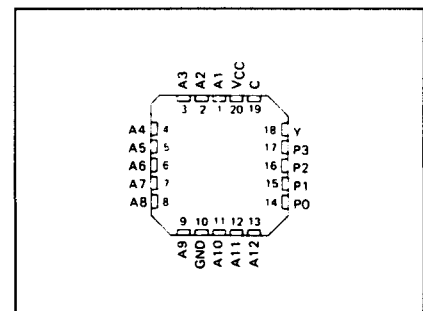
Figuur 6/3.2-528.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}							18		80 ⁴⁾	mA
I _{os}							-30		4	mA
T _{plh} ¹⁾							6		37	ns
T _{phl} ¹⁾							10		37	ns
T _{plh} ²⁾							5		21	ns
T _{phl} ²⁾							5		21	ns
T _{plh} ³⁾							3		15	ns
T _{phl} ³⁾							15		15	ns
							42			

¹⁾ P → y²⁾ A → y³⁾ C → y⁴⁾ μA

Functioneel blokschema (positieve logika).



Chip-carrier behuizing.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

'ALS680 C	INPUTS COMMON TO												'ALS680				OUTPUT Y
	P3	P2	P1	P0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	
H	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L
H	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	L
H	L	L	H	L	L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	L	H	H	L	L	L	H	H	H	H	H	H	H	H	H	L
H	L	H	L	L	L	L	L	L	H	H	H	H	H	H	H	H	L
H	L	H	L	H	L	L	L	L	L	H	H	H	H	H	H	H	L
H	L	H	H	L	L	L	L	L	L	L	H	H	H	H	H	H	L
H	L	H	H	H	L	L	L	L	L	L	L	H	H	H	H	H	L
H	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	L
H	H	L	L	H	L	L	L	L	L	L	L	L	L	H	H	H	L
H	H	L	H	L	L	L	L	L	L	L	L	L	L	L	H	H	L
H	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	H	L
H	H	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	L*
H	H	H	L	H	L	L	L	L	L	L	L	L	L	H	H	H	L*
H	H	H	H	L	L	L	L	L	L	L	L	L	L	L	H	H	L*
H	H	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L
H	All other combinations																H
L	'ALS680: Any combination																Latched

* The three shaded rows of the function table show combinations that would normally not be used in address comparator applications. The logic symbols above are not valid for these combinations in which P = 12, 13, and 14. If symbols valid for all combinations are required, starting with the fourth Exclusive-OR from the bottom, change P ≥ 9 to P = 9...11/13, 15; P ≥ 10 to P = 10...11/14/15, and P ≥ 11 to P = 11/15.

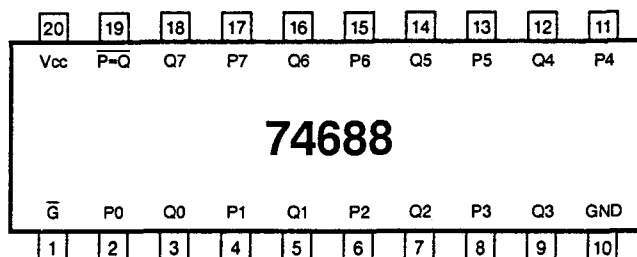
Waarheidstabel.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

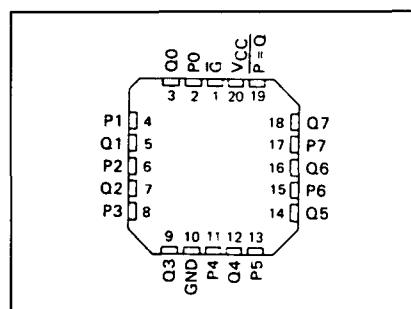
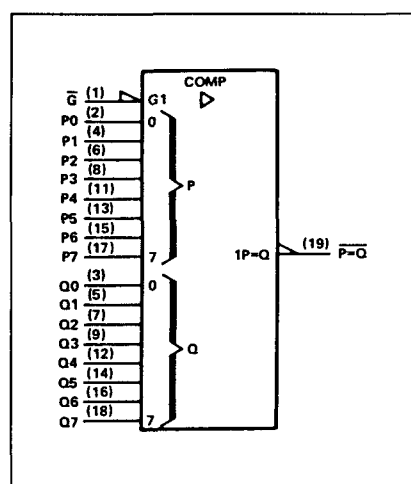
74688

8 bit identiteits comparator
met totempaal-uitgang

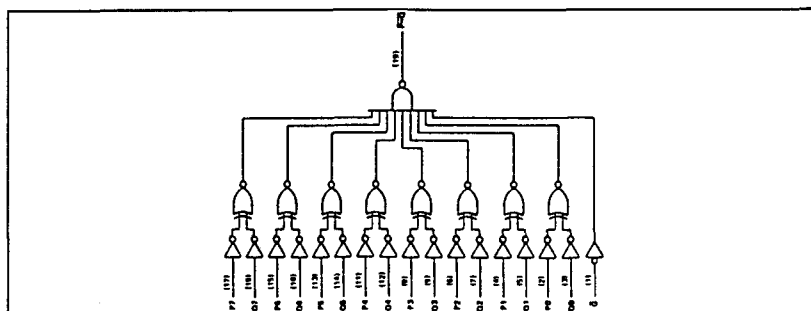
Figuur 6/3.2-688.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}					40		12		80 ⁴⁾	mA
I _{os}					-20		-30		4	mA
T _{plh} ¹⁾					18		3		30	ns
T _{phl} ¹⁾					20		5		30	ns
T _{plh} ²⁾					18		3		30	ns
T _{phl} ²⁾					20		5		30	ns
T _{plh} ³⁾					12		3		16	ns
T _{phl} ³⁾					13		5		16	ns

1) $P \rightarrow \overline{P=Q}$ 2) $Q \rightarrow \overline{P=Q}$ 3) $\overline{G} \rightarrow \overline{P=Q}$ 4) μA 

Chip-carrier behuizing.



Functioneel blokschema (positieve logica).

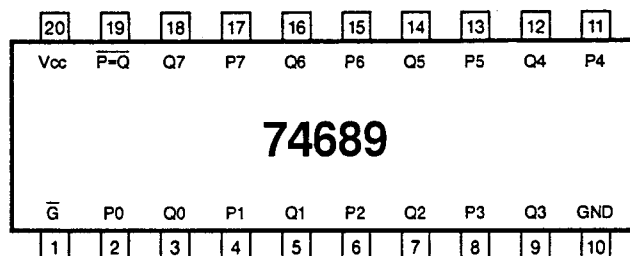
INPUTS		OUTPUT $\overline{P=Q}$
DATA P,Q	ENABLE G	
$P=Q$	L	L
$P>Q$	L	H
$P<Q$	L	H
X	H	H

Waarheidstabel.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

74689

8 bit identiteits comparator
met open-collector uitgang

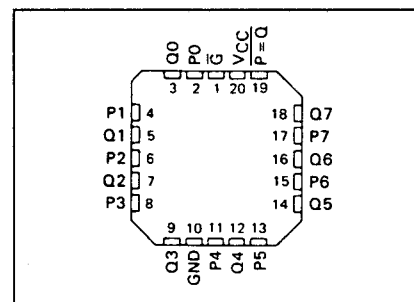
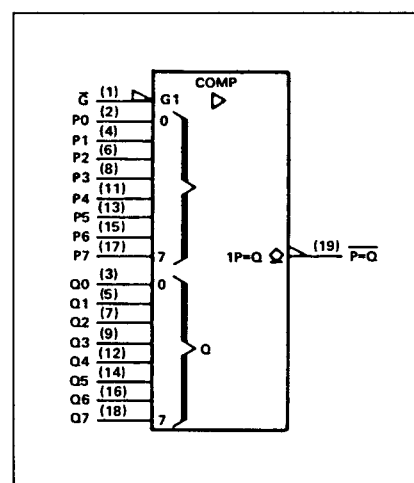


Figuur 6/3.2-689.

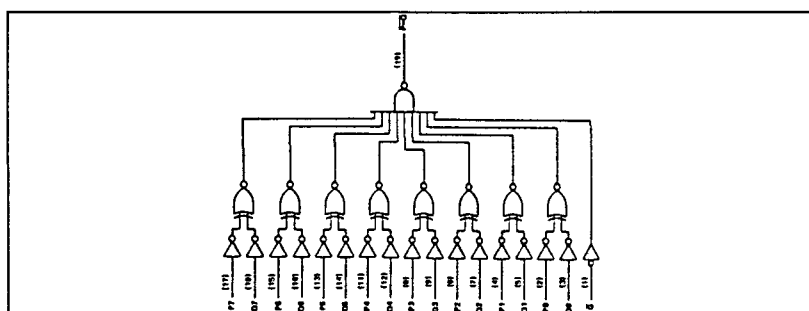
LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}					40		12			mA
I _{os}					24		24			mA
T _{plh} ¹⁾					24		10			ns
T _{phl} ¹⁾					22		5			ns
T _{plh} ²⁾					22		8			ns
T _{phl} ²⁾					19		8			ns
							25			
							25			

1) P of Q → $\overline{P=Q}$

2) $\overline{G} \rightarrow P=Q$



Chip-carrier behuizing.



Functioneel blokschema (positieve logica).

INPUTS		OUTPUT $\overline{P=Q}$
DATA	ENABLE	
P, Q	\overline{G}	
P = Q	L	L
P > Q	L	H
P < Q	L	H
X	H	H

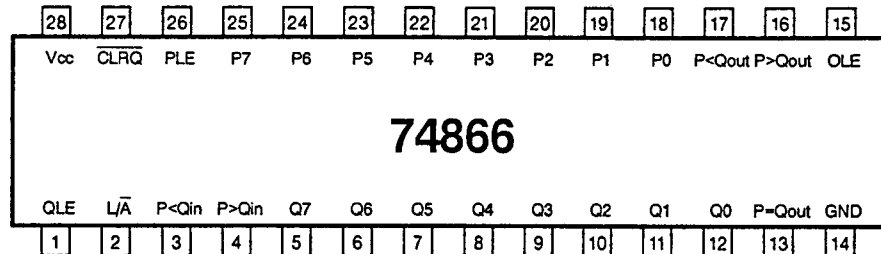
Waarheidstabel.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

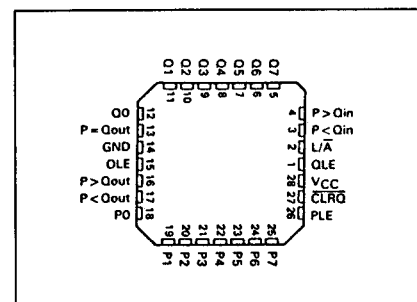
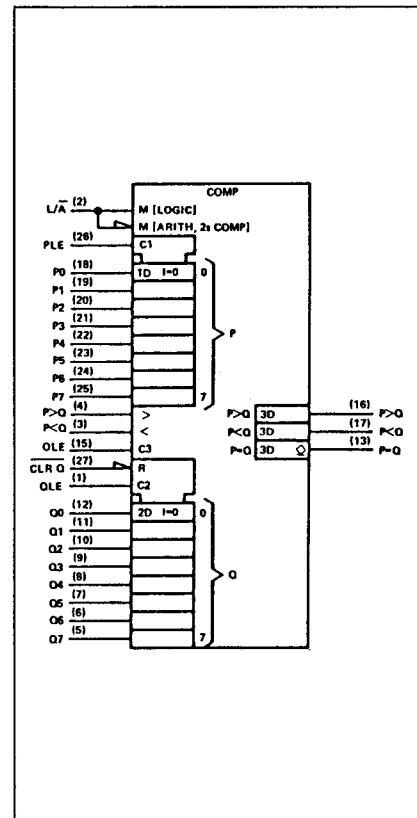
74866

8 bit magnitude
comparator

Figuur 6/3.2-866.



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}						160				mA
I _{os}						-20				mA
T _{plh} ¹⁾						8.5				ns
T _{phl} ¹⁾						7.5				ns
T _{plh} ²⁾						5				ns
T _{phl} ²⁾						5.5				ns
T _{plh} ³⁾						10				ns
T _{phl} ³⁾						9				ns
T _{plh} ⁴⁾						12				ns
T _{phl} ⁴⁾						13				ns

1) $L/\bar{A} \rightarrow P<Q$ of $P>Q$ (uitgangen)2) $P<Q$ of $P>Q \rightarrow P<Q$ of $P>Q$ (uitgangen)3) P of $Q \rightarrow P=Q$ 4) $\overline{CLRQ} \rightarrow P=Q$ 

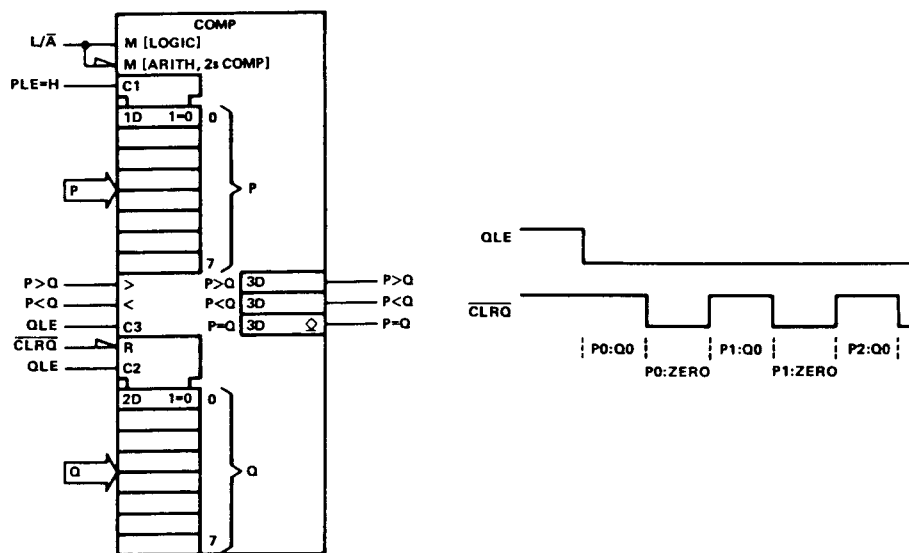
Chip-carrier behuizing.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

COMPARISON	L/ \bar{A}	DATA INPUTS P0-P7, Q0-Q7	INPUTS		OUTPUTS		
			P > Q	P < Q	P > Q	P < Q	P = Q
Logical	H	P > Q	X	X	H	L	L
Logical	H	P < Q	X	X	L	H	L
Logical	H	P = Q	L	L	L	L	H
Logical	H	P = Q	L	H	L	H	L
Logical	H	P = Q	H	L	H	L	L
Logical	H	P = Q	H	H	H	H	L
Arithmetic	L	P AG Q	X	X	H	L	L
Arithmetic	L	Q AG P	X	X	L	H	L
Arithmetic	L	P = Q	L	L	L	L	H
Arithmetic	L	P = Q	L	H	L	H	L
Arithmetic	L	P = Q	H	L	H	L	L
Arithmetic	L	P = Q	H	H	H	H	L

AG = arithmetically greater than

Waarheidstabel.



-MAGNITUDE COMPARISONS COMBINED WITH QUICK COMPARISONS TO ZERO (RANGE VERIFICATIONS)

Toepassingsvoorbeeld.

gebruik van $\overline{\text{CLRQ}}$ voor dubbele vergelijking:

- als $\overline{\text{CLRQ}}$ = LAAG wordt de P-term vergeleken met nul
- als $\overline{\text{CLRQ}}$ = HOOG wordt de P-term vergeleken met de Q-term

Vergelijking van grotere woordlenten is mogelijk door telkens de P>Q en P<Q uitgangen van een trap te verbinden met de P>Q en P<Q ingangen van de aansluitend hogere trap. De open-collector P=Q uitgangen kunnen als "wired-AND" worden geschakeld.

3.2 Magnitude comparatoren 74xx-serie TTL en HC

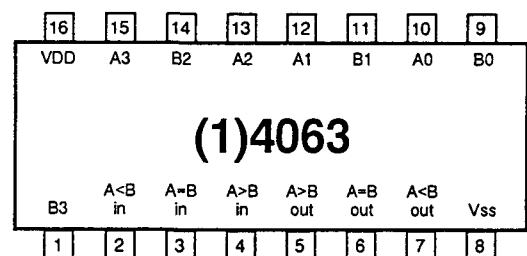
6/3.3

Magnitude comparatoren
(1)4xxx-serie CMOS

(1)4063

4 bit magnitude
comparator

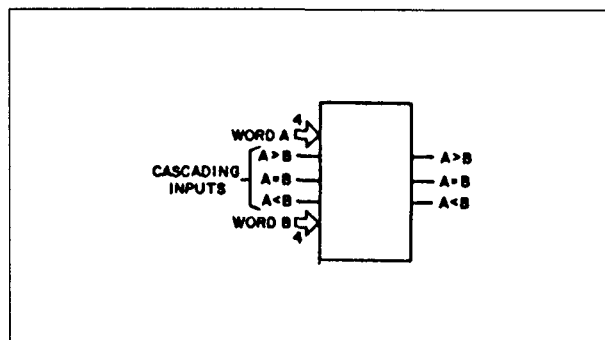
Figuur 6/3.3-63.



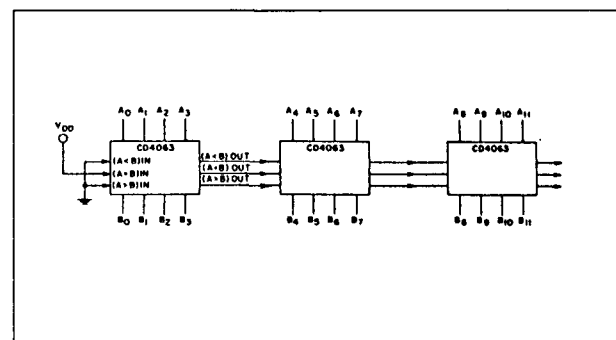
INPUTS								OUTPUTS		
COMPARING				CASCADING				A < B	A = B	A > B
A3, B3	A2, B2	A1, B1	A0, B0	A < B	A = B	A > B		A < B	A = B	A > B
A3 > B3	X	X	X	X	X	X	0	0	1	
A3 = B3	A2 > B2	X	X	X	X	X	0	0	1	
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	0	0	1	
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	0	0	1	
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	0	1	0	0	1	
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	1	0	0	1	0	
A3 = B3	A2 = B2	A1 = B1	A0 = B0	1	0	0	1	0	0	
A3 < B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	1	0	0	
A3 < B3	A2 = B2	A1 < B1	X	X	X	X	1	0	0	
A3 < B3	A2 < B2	X	X	X	X	X	1	0	0	
A3 < B3	X	X	X	X	X	X	1	0	0	

X = Don't Care Logic 1 = High Level Logic 0 = Low Level

Waarheidstabel.



Functioneel schema.



Toepassingsvoorbeeld: 12 bit comparator.

o.a. leverbaar:

CD 4063 B, MSM 4063, μ PD 4063 B

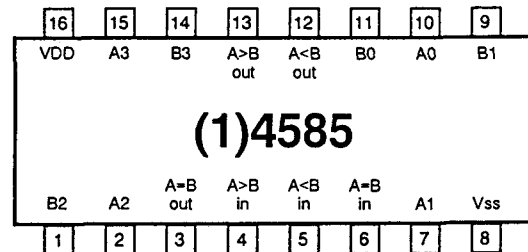
3.3 Magnitude comparatoren (1)4xxx-serie CMOS

(1)4585

4 bit

magnitude-comparator

Figuur 6/3.3-85.



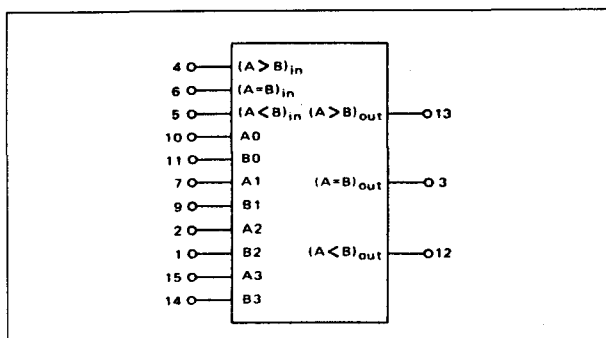
comparing inputs				cascading inputs			outputs		
A ₃ , B ₃	A ₂ , B ₂	A ₁ , B ₁	A ₀ , B ₀	I _A > B	I _A < B	I _A = B	O _A > B	O _A < B	O _A = B
A ₃ > B ₃	X	X	X	H	X	X	H	L	L
A ₃ < B ₃	X	X	X	X	X	X	L	H	L
A ₃ = B ₃	A ₂ > B ₂	X	X	H	X	X	H	L	L
A ₃ = B ₃	A ₂ < B ₂	X	X	X	X	X	L	H	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ > B ₁	X	H	X	X	H	L	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ < B ₁	X	X	X	X	L	H	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ > B ₀	H	X	X	H	L	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ < B ₀	X	X	X	L	H	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	X	L	H	L	L	H
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	H	L	L	H	L	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	X	H	L	L	H	L
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	X	H	H	L	H	H
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	L	L	L	L	L	L

H = HIGH state (the more positive voltage)

L = LOW state (the less positive voltage)

X = state is immaterial

Waarheidstabel.

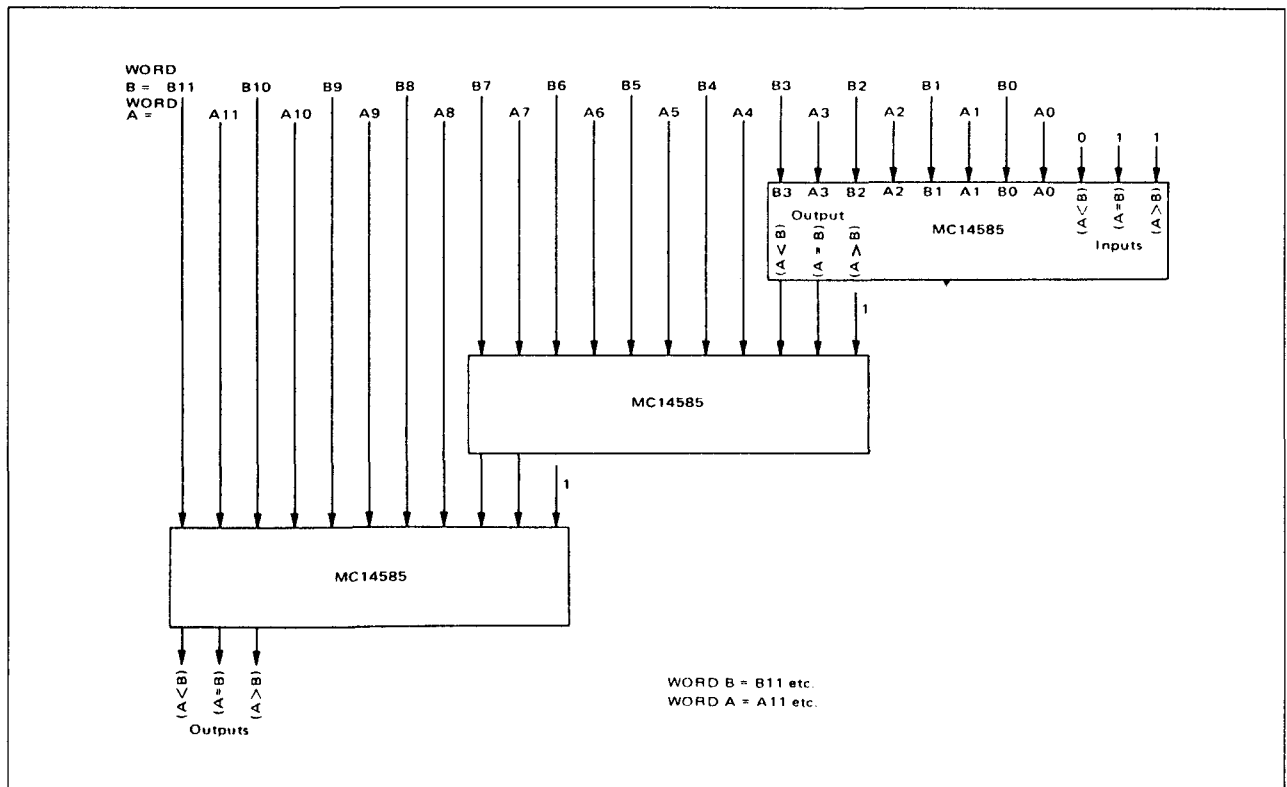


Blok-schema.

o.a. leverbaar:

MC 14585 A/B/C, HEF 4585 B, CD 4585 B

3.3 Magnitude comparatoren (1)4xxx-serie CMOS



Toepassingsvoorbeeld: 12 bit comparator.

3.3 Magnitude comparatoren (1)4xxx-serie CMOS

6/4

Binaire multipliers

Inhoud

6/4.1 Achtergrond-informatie
(aanvulling 29)**6/4.2 Binaire multipliers 74xx-serie TTL en HC**
(aanvulling 17)

7497	synchrone 6-bit binaire rate multiplier
74167	synchrone decade rate multiplier
74261	2-bit x 4-bit parallel binaire multiplier
74274	4-bit x 4-bit binaire multiplier, 3-state
74275	7-bit slice Wallace trees, 3-state
74284	4-bit x 4-bit parallel binaire multiplier
74285	4-bit x 4-bit parallel binaire multiplier
74384	8-bit x 1-bit two's complement multiplier

6/4.3 Binaire multipliers (1)4xxx-serie CMOS
(aanvulling 29)

(1)4089	binaire rate multiplier
(1)4527	BCD rate multiplier
(1)4554	2-bit x 2-bit binaire multiplier

6/4.1

Achtergrond-informatie

Inleiding

De binaire multipliers kunnen worden opgesplitst in twee groepen:

- de binaire rate multipliers;
- de "echte" binaire multipliers.

Rate multipliers

Dit zijn eigenlijk delers waarbij de frequentie van een clocksignaal door een instelbaar getal wordt gedeeld. Als voorbeeld wordt een BCD Rate Multiplier van het type 4527 besproken. In het tijddiagram van figuur 6/4.1-1 ziet men dat van elke 10 clockpulsen er n worden doorgegeven aan de uitgang.

Hierbij is n het ingangsgetal. Is bijvoorbeeld $n = 6$ (BCD 0110), dan komen er per 10 clockpulsen 6 aan de uitgang.

Rate Multipliers kunnen worden toegepast voor rekenkundige bewerkingen, zoals vermenigvuldigen en delen, bij analoog/digitaal- en digitaal/analoog omzetting en als frequentiedeler. Bij het laatste moet rekening worden gehouden met het feit dat de uitgangspulsen meestal niet gelijkmatig over de tijd zijn verdeeld.

Binaire multipliers

De werking hiervan is wat ingewikkelder. Vandaar wordt eerst het binaire vermenigvuldigen behandeld. De hierbij geldende regels zijn geschetst in figuur 6/4.1-2. Wanneer men zich daaraan houdt is er eigenlijk geen verschil tussen binair en decimaal vermenigvuldigen. In het voorbeeld van figuur 6/4.1-3 is te zien dat de methode identiek is. Een binair vermenigvuldigtal 1101 (= decimaal

13) wordt vermenigvuldigd met de vermenigvuldiger 1010 (= decimaal 10). Eerst vindt vermenigvuldiging plaats van de laagste bit (1-en) van de vermenigvuldiger en het vermenigvuldigtal. Het resultaat daarvan is 0000. Daarna is het de beurt van het 2-bit van de vermenigvuldiger, met als tussenresultaat 1101. Merk op dat dit tussenprodukt één plaats naar links is verschoven ten opzichte van het eerste tussenprodukt. Op deze wijze wordt de gehele vermenigvuldiging uitgevoerd tot het eindprodukt 10000110 is bereikt (= decimaal 130).

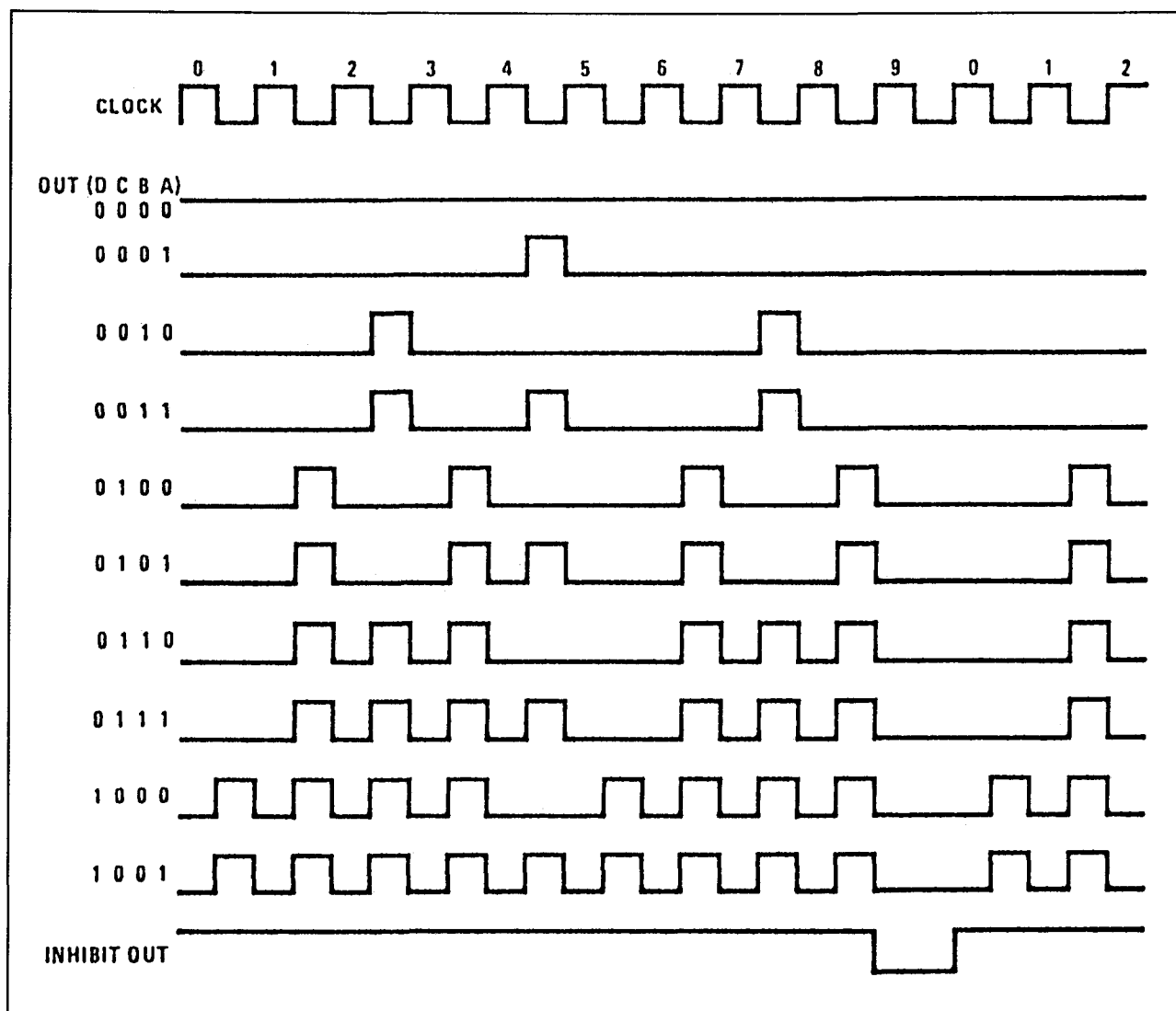
In figuur 6/4.1-4 is de gehele procedure nogmaals te zien, maar dan aangevuld met het proces dat door de binaire logika wordt gebruikt bij de "tel-op-en-schuif" methode (add-and-shift). De regels 1 en 2 bevatten de beide te vermenigvuldigen binaire getallen. Op regel 3 ziet men dat de 1-en vermenigvuldiger (die gelijk is aan 0) wordt vermenigvuldigd met het vermenigvuldigtal. Het eerste partiële produkt is 0000. Regel 4 toont een "schuif-links" voor het volgende partiële produkt (of een "schuif-rechts" voor het eerste partiële produkt).

Op regel 5 wordt de 2-en bit (=1) vermenigvuldigd met het vermenigvuldigtal. Dit levert 1101 op en wordt aangevuld met de meest rechtse 0.

Het tweede partiële produkt bedraagt dus 11010.

Op regel 6 wordt het eerste partiële produkt bij het tweede opgeteld met 11010 als resultaat.

4.1 Achtergrond-informatie



Figuur 6/4.1-1: Tijddiagram van een BCD Rate Multiplier.

0	0	1	1
$\times 0$	$\times 1$	$\times 0$	$\times 1$
0	0	0	1

Figuur 6/4.1-2: Regels voor binair vermenigvuldigen.

Decimal	Binary	
13	1101	Multiplicand
$\times 10$	$\times 1010$	Multiplier
00	0000	1st partial product
13	1101	2nd partial product
130	0000	3rd partial product
	1101	4th partial product
	1000010	Product

Figuur 6/4.1-3: Een voorbeeld van binair vermenigvuldigen.

4.1 Achtergrond-informatie

1	Multiplicand	1101	
2	Multiplier	$\times 1010$	
3	1st partial product	0000	1s multiplier bit = 0. Write 0000.
4		----	Shift left.
5	2nd partial product	11010	2s multiplier bit = 1. Copy multiplicand: 1101.
6		11010	Add 1st and 2nd partial products.
7		----00	Shift left.
8	3rd partial product	000000	4s multiplier bit = 0. Write 0000.
9		011010	Add 1st and 2nd plus 3rd partial products.
10		----000	Shift left.
11	4th partial product	1101000	8s multiplier bit = 1. Copy multiplicand: 1101.
12	Product	1000010	Add 1st, 2nd, and 3rd plus 4th partial products.

Figuur 6/4.1-4: Hetzelfde voorbeeld als in figuur 6/4.1-3, aangevuld met de procedure die door logische schakelingen wordt gebruikt voor binaire vermenigvuldiging.

Regel 7 toont nogmaals een "schuif-links". De 4-en bit van de vermenigvuldiger is gelijk aan 0, zodat het derde partiële produkt 0000 wordt. Op regel 8 wordt hier 00 aan toegevoegd en wordt het volledige derde partiële produkt 00000. Op regel 9 worden de regels 6 en 8 bij elkaar opgeteld, enzovoorts.

Deze "tel-op-en-schuif methode" kan met digitale schakelingen worden uitgevoerd.

Wanneer men het binaire vermenigvuldigingsproces nauwkeurig bekijkt, dan vallen drie belangrijke dingen op:

- Partiële produkten zijn 0000 als de vermenigvuldiger-bit gelijk aan 0 is, of gelijk aan het vermenigvuldigtal als de vermenigvuldiger-bit gelijk aan 1 is.
- Het eindresultaat kan tweemaal zo lang zijn als het vermenigvuldigtal.
- Bij het optellen wordt het eerste partiële produkt één plaats naar rechts geschoven ten opzichte van het tweede partiële produkt.

Binaire multiplier

Deze feiten vormen de basis van een binaire vermenigvuldiger zoals geschetst is in figuur 6/4.1-5. Deze schakeling bevat een 5-bit parallel opteller (adder) met een add/do-not-add besturing (wel/niet optellen).

Het register dat het vermenigvuldigtal (multiplicand) bevat is een 4-bit schuifregister (links boven).

Rechts onder bevinden zich een 5-bit accumulator-register en een 4-bit vermenigvuldiger-register (multiplier).

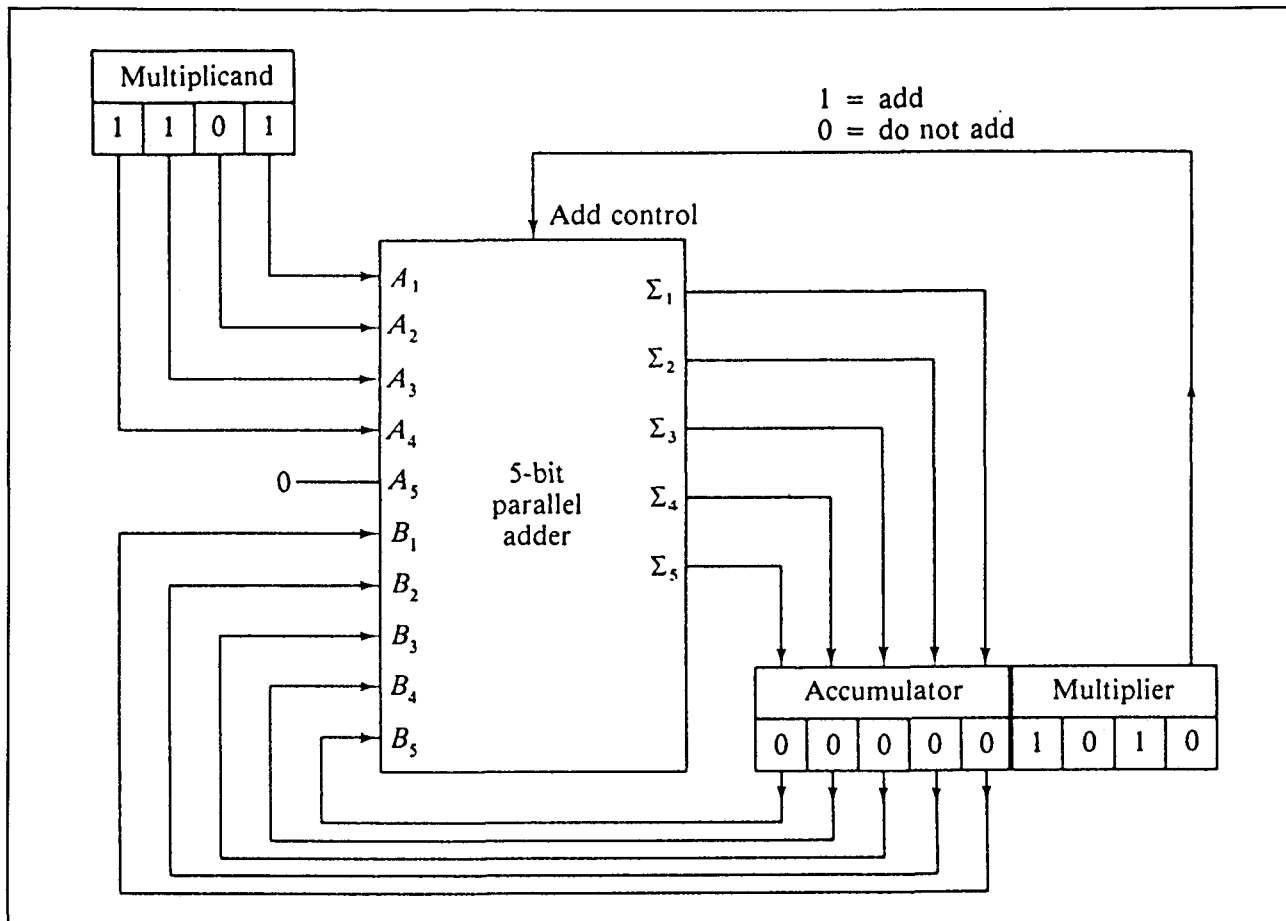
Net als in het voorbeeld van figuur 6/4.1-4 wordt nu de multiplicand geladen met 1101 en de multiplier met 1010. De accumulator wordt leeggemaakt (00000). Deze situatie komt overeen met stap 1 in figuur 6/4.1-6. Stap 2 is het optellen uit de optel-en-schuif procedure.

De minst belangrijke bit (LSB) van de multiplier wordt toegevoerd aan de add-besturing van de 5-bit adder. In dit geval is het een 0, zodat geen optelling wordt uitgevoerd. Alle registers blijven gelijk. Deze stap komt overeen met regel 3 in figuur 6/4.1-4. Het eerste partiële produkt (0000) komt in het accumulator-register.

Bij stap 3 worden de inhouden van de accumulator en de multiplier één plaats naar rechts geschoven. Hierbij wordt van links een 0 ingebracht en gaat aan de rechterzijde een 0 verloren.

Zoals aan het donkere gedeelte te zien is, blijven slechts drie bits van de originele waarde in de multiplier over.

4.1 Achtergrond-informatie



Figuur 6/4.1-5: Een binaire vermenigvuldiger.

Bij stap 4 wordt weer opgeteld. De LSB in het multiplier-register laat de 5-bit adder optellen. De 00000 uit de accumulator wordt nu bij de 01101 uit het multiplicand-register opgeteld. Het resultaat (01101) wordt opgeborgen in de accumulator. De som van de partiële produkten op dit moment (011010) is te zien in het lichte gedeelte van zowel de accumulator als de multiplier. Dit komt overeen met regel 6 van figuur 6/4.1-4. Bij stap 5 wordt weer naar rechts geschoven. Van links komt een 0 binnen en naar rechts gaat nu een 1 verloren. Bij stap 6 gaat het optellen nu niet door, omdat de LSB van de multiplier nu een 0 is en blijven de registers onveranderd.

Bij stap 7 wordt nogmaals naar rechts geschoven, waardoor van links een 0 binnen-

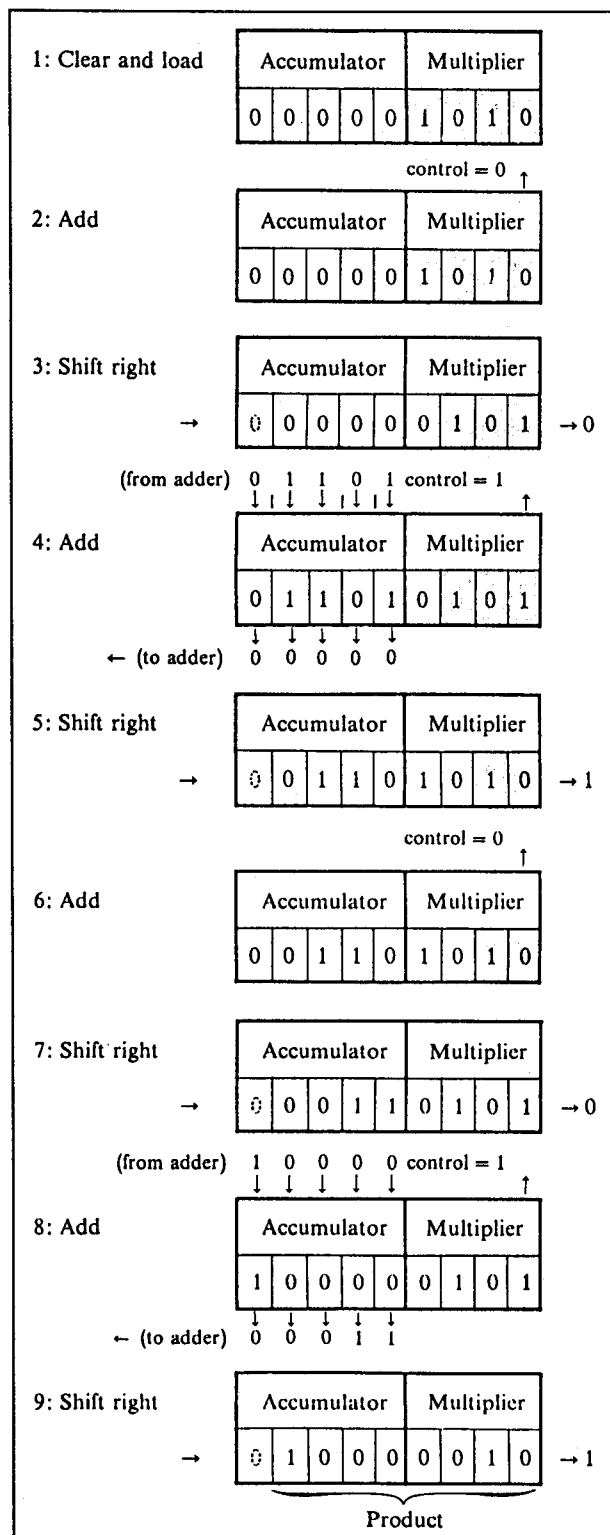
komt en naar rechts een 0 verloren gaat. Bij stap 8 wordt nu wel opgeteld omdat de LSB een 1 is. De inhoud van de accumulator (00011) wordt op de B-zijde van de adder gezet en opgeteld bij de inhoud van de multiplicand die onveranderd 1101 blijft.

De som (10000) wordt in de accumulator geladen en bij stap 9 wordt voor de laatste maal naar rechts geschoven. Het eindresultaat bevindt zich nu in de accumulator- plus de multiplier-registers.

Vermenigvuldigen met herhaald optellen

De tel-op-en-schuif procedure kan met logische schakelingen worden gebouwd of als computerprogramma worden geschreven. Een andere methode van vermenigvuldigen is natuurlijk het herhaald optellen.

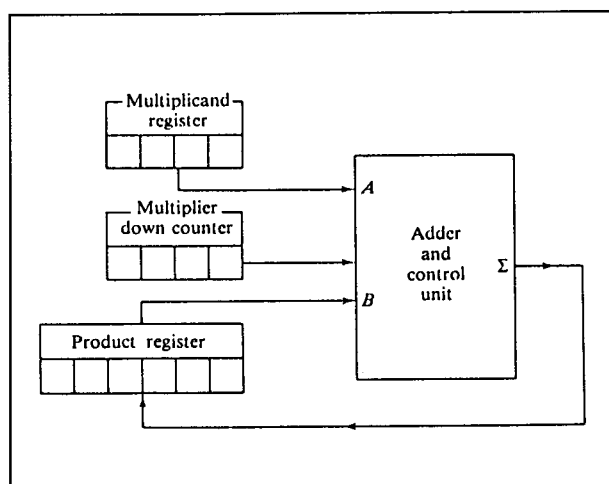
4.1 Achtergrond-informatie



Figuur 6/4.1-6: Werking van de binaire vermenigvuldiger.

Wanneer bijvoorbeeld 6×5 moet worden uitgerekend, kan $6 + 6 + 6 + 6 + 6 = 30$ worden uitgevoerd.

Het spreekt vanzelf dat één van beide getallen dan in een teller wordt geladen die na elke optelling met één wordt verlaagd (zie figuur 6/4.1-7).



Figuur 6/4.1-7: Vermenigvuldigen door middel van herhaald optellen.

4.1 Achtergrond-informatie

6/4.2

Binaire multipliers
74xx-serie TTL en HC

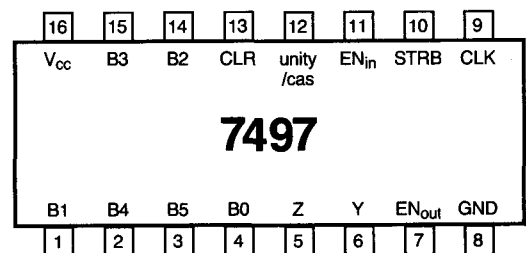
7497

Synchrone 6-bit binaire rate
multiplier

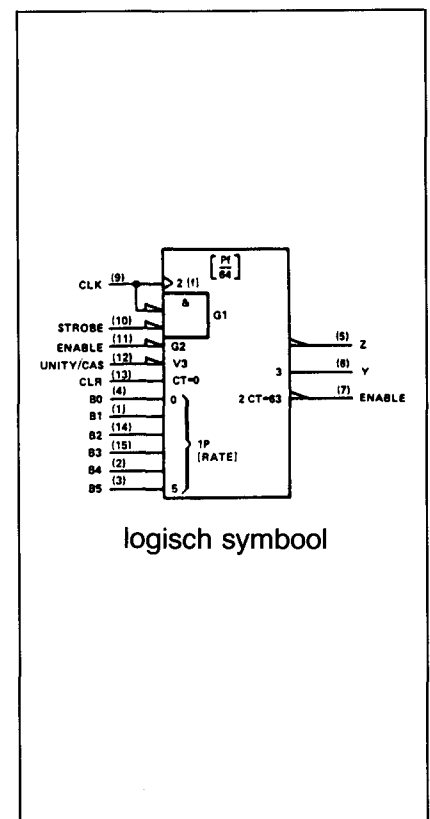
$$f_{out} = \frac{M \cdot f_{in}}{64}$$

Figuur 6/4.2-97

$$(M = F \cdot 2^5 + E \cdot 2^4 + D \cdot 2^3 + C \cdot 2^2 + B \cdot 2^1 + A \cdot 2^0)$$



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc} ^H L	58 80									mA
I _{os}	-18 -55									mA
T _{plh} / T _{phl} ¹⁾	13/ 14									ns
T _{plh} / T _{phl} ²⁾	12/ 15									ns
T _{plh} / T _{phl} ³⁾	26/ 20									ns
T _{plh} / T _{phl} ⁴⁾	12/ 17									ns
T _{plh} / T _{phl} ⁵⁾	6/ 9									ns
T _{plh} / T _{phl} ⁶⁾	15/ 15									ns
T _{plh} / T _{phl} ⁷⁾	9/ 6									ns
f _{max}	32									MHz



- 1) Enable in → Enable Out
 2) Strobe → Z
 3) Clock → Y
 4) Clock → Z
 5) Rate (B0 t/m B5) → Z
 6) Rate (B0 t/m B5) → Y
 7) Unity/Cascade → Y

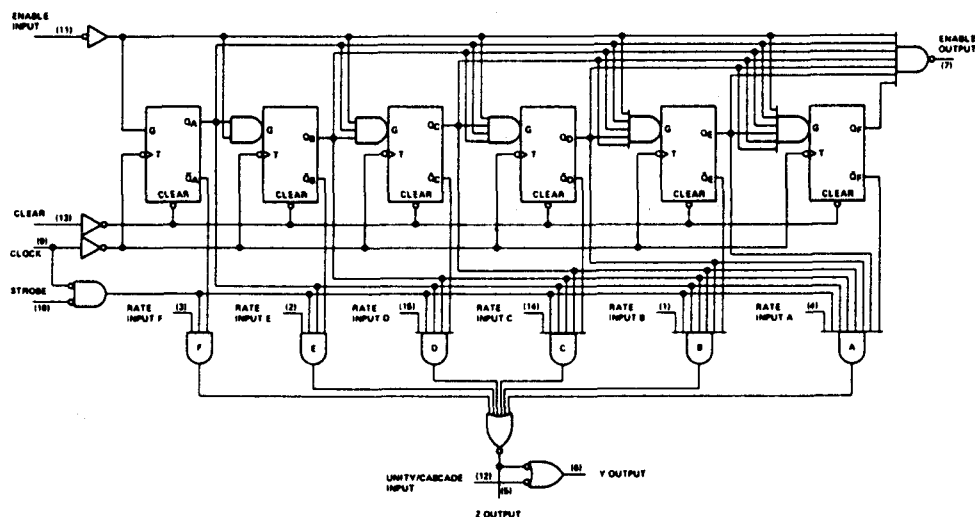
B0 = A
 B1 = B
 B2 = C
 B3 = D
 B4 = E
 B5 = F

4.2 74xx-serie TTL en HC

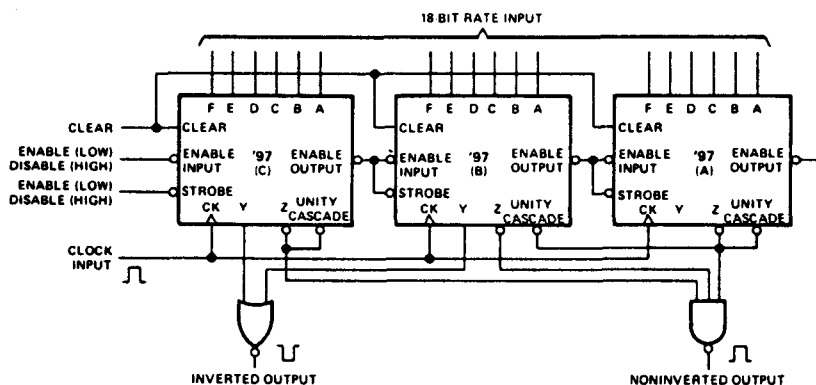
INPUTS										OUTPUTS			NOTES	
CLEAR	ENABLE	STROBE	BINARY RATE F E D C B A						NUMBER OF CLOCK PULSES	UNITY/ CASCADE	LOGIC LEVEL OR NUMBER OF PULSES			
											Y	Z		ENABLE
H	X	H	X	X	X	X	X	X	H	L	H	H	B	
L	L	L	L	L	L	L	L	L	64	H	L	H	1	C
L	L	L	L	L	L	L	L	H	64	H	1	1	1	C
L	L	L	L	L	L	L	H	L	64	H	2	2	1	C
L	L	L	L	L	L	H	L	L	64	H	4	4	1	C
L	L	L	L	L	H	L	L	L	64	H	8	8	1	C
L	L	L	L	H	L	L	L	L	64	H	16	16	1	C
L	L	L	H	L	L	L	L	L	64	H	32	32	1	C
L	L	L	H	H	H	H	H	H	64	H	63	63	1	C
L	L	L	L	H	H	H	H	H	64	L	H	63	1	D
L	L	L	L	H	L	H	L	L	64	H	40	40	1	E

- NOTES: A. H = high level, L = low level, X = irrelevant. All remaining entries are numeric counts.
 B. This is a simplified illustration of the clear function. The states of clock and strobe can affect the logic level of Y and Z. A low unity/cascade will cause output Y to remain high.
 C. Each rate illustrated assumes a constant value at rate inputs; however, these illustrations in no way prohibit variable rate inputs.
 D. Unity/cascade is used to inhibit output Y.
 E. $f_{out} = \frac{M \cdot f_{in}}{64} = \frac{(8 \times 32) f_{in}}{64} = \frac{40 f_{in}}{64} = 0.625 f_{in}$

waarheidstabel



functioneel blokschema

toepassingsvoorbeeld
(uitgebreid tot 18 bits)

4.2 74xx-serie TTL en HC

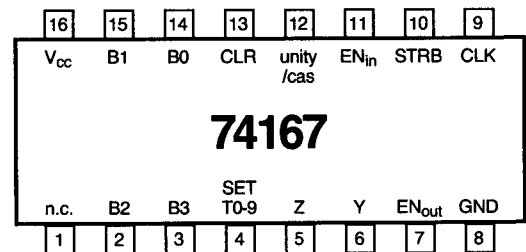
74167

synchrone decade rate multiplier

$$f_{out} = \frac{M \cdot f_{in}}{10}$$

(M = D.2³ + C.2² + B.2¹ + A.2⁰) (0 t/m 9)

Figuur 6/4.2-167



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc} ^H L	43 65									mA
I _{os}	-18 -55									mA
T _{plh} / T _{phl} ¹⁾	13/ 14									ns
T _{plh} / T _{phl} ²⁾	12/ 15									ns
T _{plh} / T _{phl} ³⁾	26/ 20									ns
T _{plh} / T _{phl} ⁴⁾	12/ 17									ns
T _{plh} / T _{phl} ⁵⁾	9/ 6									ns
T _{plh} / T _{phl} ⁶⁾	15/ 15									ns
T _{plh} / T _{phl} ⁷⁾	9/ 6									ns
f _{max}	32									MHz

1) Enable in → Enable out

2) Strobe → Z

3) Clock → Y

4) Clock → Z

5) Rate (A,B,C,D) → Z

6) Rate (A,B,C,D) → Y

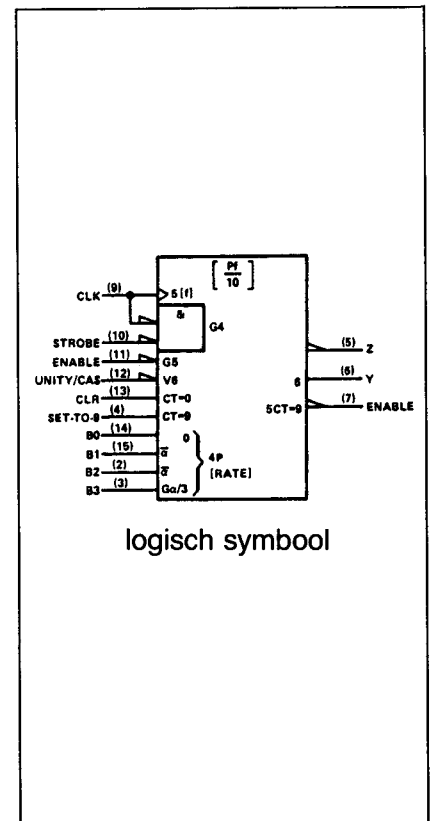
7) Unity/Cascade → Y

B0 = A

B1 = B

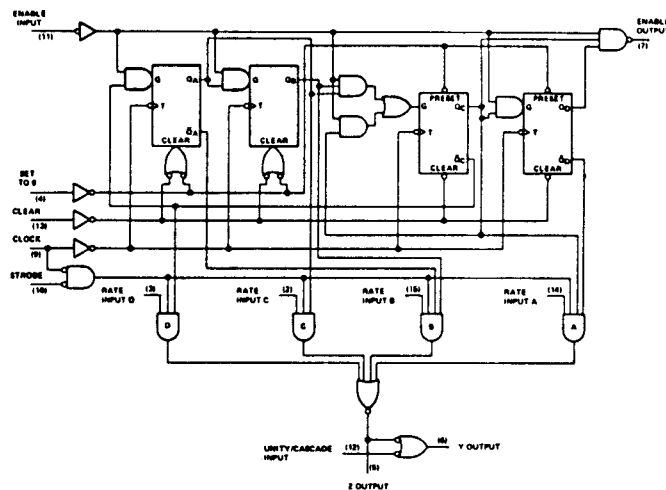
B2 = C

B3 = D



Deel 6: Digitale rekenkundige schakelingen

4.2 74xx-serie TTL en HC



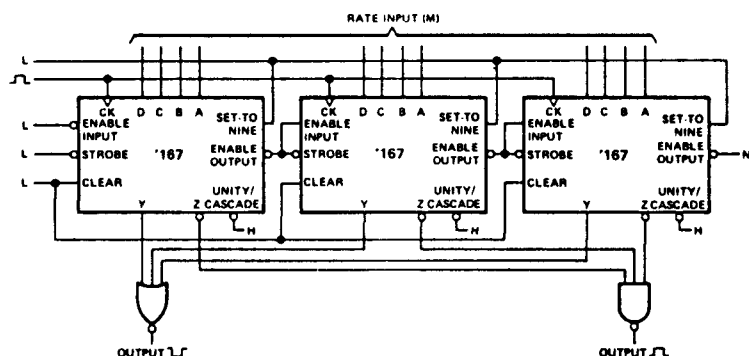
functioneel blokschema

STATE AND/OR RATE FUNCTION TABLE (See Note A)

INPUTS								OUTPUTS				NOTES
CLEAR	ENABLE	STROBE	BCD RATE D C B A				NUMBER OF CLOCK PULSES	UNITY/ CASCADE	LOGIC LEVEL OR NUMBER OF PULSES			
									Y	Z	ENABLE	
H	X	H	X	X	X	X	X	H	L	H	H	B
L	L	L	L	L	L	L	10	H	L	H	1	C
L	L	L	L	L	L	L	10	H	1	1	1	C
L	L	L	L	L	L	H	10	H	2	2	1	C
L	L	L	L	L	L	H	10	H	3	3	1	C
L	L	L	L	L	H	L	10	H	4	4	1	C
L	L	L	L	L	H	L	10	H	5	5	1	C
L	L	L	L	L	H	H	10	H	6	6	1	C
L	L	L	L	L	H	H	10	H	7	7	1	C
L	L	L	L	H	L	L	10	H	8	8	1	C
L	L	L	L	H	L	H	10	H	9	9	1	C
L	L	L	H	L	H	L	10	H	8	8	1	C, D
L	L	L	H	L	H	L	10	H	9	9	1	C, D
L	L	L	H	H	L	L	10	H	8	8	1	C, D
L	L	L	H	H	L	H	10	H	9	9	1	C, D
L	L	L	H	H	H	L	10	H	8	8	1	C, D
L	L	L	H	H	H	H	10	H	9	9	1	C, D
L	L	L	H	L	L	H	10	L	H	9	1	E

- NOTES: A. H = high level, L = low level, X = irrelevant. All remaining entries are numeric counts.
 B. This is a simplified illustration of the clear function. The states of clock and strobe can affect the logic level of Y and Z. A low unity/cascade will cause output Y to remain high.
 C. Each rate illustrated assumes a constant value at rate inputs; however, these illustrations in no way prohibit variable-rate inputs.
 D. These input conditions exceed the range of the decimal rate inputs.
 E. Unity/cascade can be used to inhibit output Y.

waarheidstabel

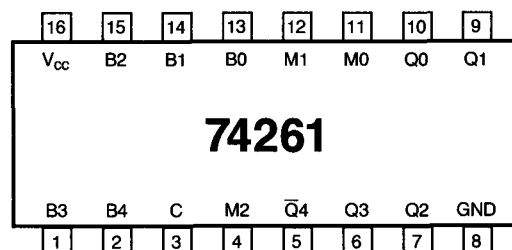
toepassingsvoorbeeld
0,999 t/m 999

4.2 74xx-serie TTL en HC

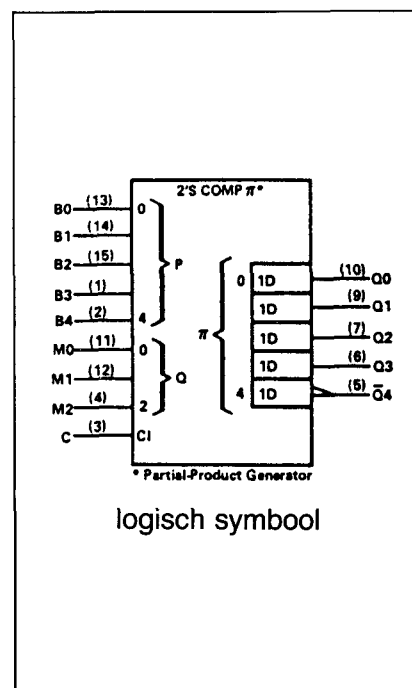
74261

2-bit by 4-bit parallelle binaire multiplier

Figuur 6/4.2-261



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
I _{cc}					20					mA
I _{os}					-20 -100					mA
T _{plh} ¹⁾					22					ns
T _{phl} ¹⁾					20					ns
T _{plh} ²⁾					25					ns
T _{phl} ²⁾					22					ns
T _{plh} ³⁾					27					ns
T _{phl} ³⁾					24					ns



- 1) Enable G → Q
 2) M input → Q
 3) B input → Q

FUNCTION TABLE

LATCH CONTROL G	INPUTS			OUTPUTS					
	MULTIPLIER			Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	
L	X	X	X	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	
H	L	L	L	H	L	L	L	L	
H	L	L	H	B ₄	B ₄	B ₃	B ₂	B ₁	
H	L	H	L	B ₄	B ₄	B ₃	B ₂	B ₁	
H	L	H	H	B ₄	B ₃	B ₂	B ₁	B ₀	
H	H	L	L	B ₄	B ₃	B ₂	B ₁	B ₀	
H	H	L	H	B ₄	B ₄	B ₃	B ₂	B ₁	
H	H	H	L	B ₄	B ₄	B ₃	B ₂	B ₁	
H	H	H	H	H	L	L	L	L	

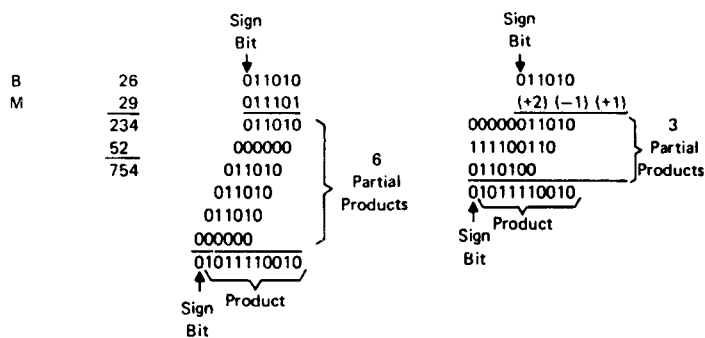
H = high level, L = low level, X = irrelevant
 Q₄...Q₀ = The logic level of the same output before the high-to-low transition of G.
 B₄...B₀ = The logic level of the indicated multiplicand (B) input.

waarheidstabel

DECIMAL

BINARY

2-BIT-AT-A-TIME BINARY

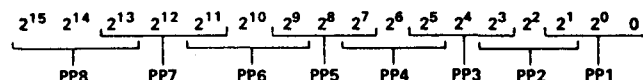


Voorbeeld: vermenigvuldiging van de getallen 26 (vermenigvuldigd B) en 29 (vermenigvuldiger M) decimaal, binair en 'twee bits tegelijk' binair:

4.2 74xx-serie TTL en HC

Partial-product-generation regels zijn:

- 1) Neem twee bits van de vermenigvuldiger M plus het aansluitend lagere bit. Voor het eerste partial-product (PP1) is het aansluitend lager bit nul:



- 2) Genereer PPi volgens tabel:

MULTIPLIER BITS FROM STEP 1			OPERATOR SYMBOL	TO OBTAIN PARTIAL PRODUCT
2^{2i-1}	2^{2i-2}	2^{2i-3}		
0	0	0	0	Replace multiplicand by zero
0	0	1	+1 B	Copy multiplicand
0	1	0	+1 B	Copy multiplicand
0	1	1	+2 B	Shift multiplicand left one bit
1	0	0	-2 B	Shift two's complement of multiplicand left one bit
1	0	1	-1 B	Replace multiplicand by two's complement
1	1	0	-1 B	Replace multiplicand by two's complement
1	1	1	0	Replace multiplicand by zero

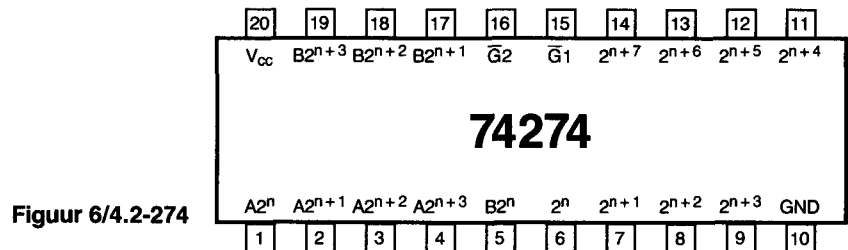
- 3) Geef de partial products 'gewicht' door elke twee plaatsen links van het volgende minst-significante produkt te indexeren.
- 4) Verleng het meest-significante bit van het partial product tot de plaats van het teken-bit van het eindprodukt.

Voorbeeld van de algoritme:

M = 29 = 011101	Operator Symbol	B = 26 = 011010
<pre> 011101 010 110 011 </pre>	<pre> +1 B -1 B +2 B </pre>	<pre> 00000011010 111100110 0110100 </pre>

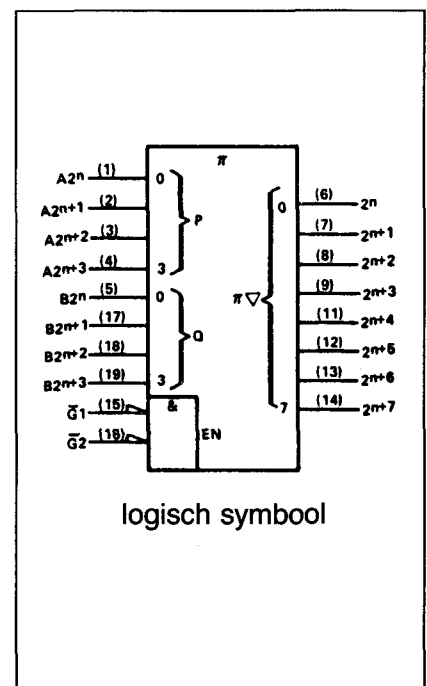
4.2 74xx-serie TTL en HC

74274

4-bit by 4-bit binaire multiplier met
3-state uitgangen

Figuur 6/4.2-274

LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
Icc				105						mA
Ios				-30 -100						mA
Tplh ¹⁾				50						ns
Tphi ¹⁾				50						ns
Tpzh ²⁾				15						ns
Tpzi ²⁾				15						ns
Tphz ²⁾				10						ns
Tplz ²⁾				10						ns

¹⁾ A of B → alle uitgangen²⁾ G1 of G2 → alle uitgangen

De 74S274 is een 4-bit by 4-bit parallele vermenigvuldiger. Voor het verkrijgen van een 8-bit product zijn geen extra componenten nodig. Voor grotere woordlengten dan 4 bits kunnen een aantal 74S274 multipliers worden gecombineerd voor het genereren van sub-multipelen partiële produkten. Deze 'partial products' kunnen vervolgens in Wallace trees worden gecombineerd ter verkrijging van het eindproduct. De figuren A en B behoren dan samen met figuur C van de 74S275 te worden gebruikt.

4.2 74xx-serie TTL en HC

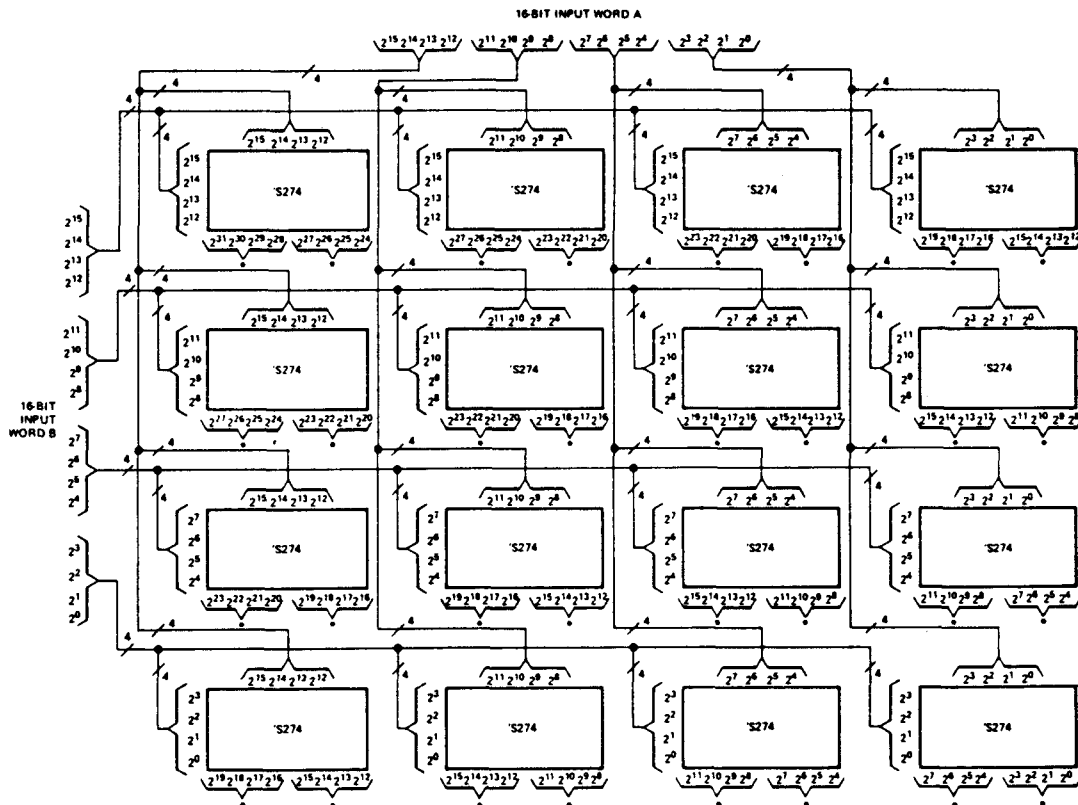


Fig. A

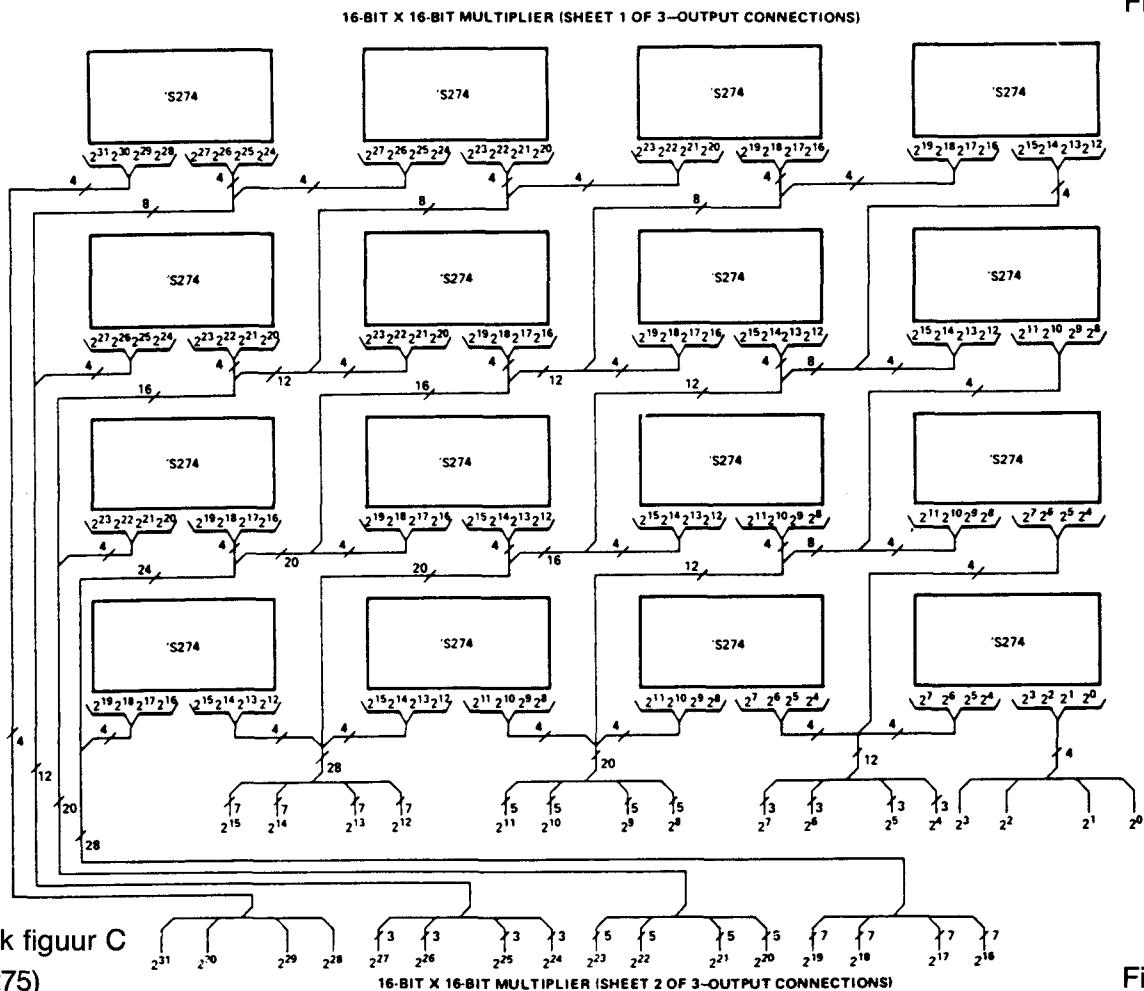


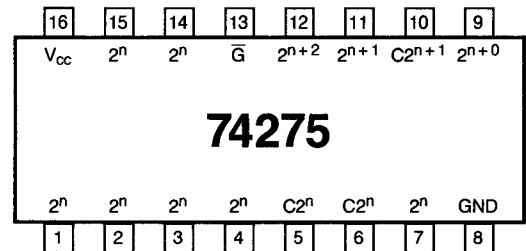
Fig. B

4.2 74xx-serie TTL en HC

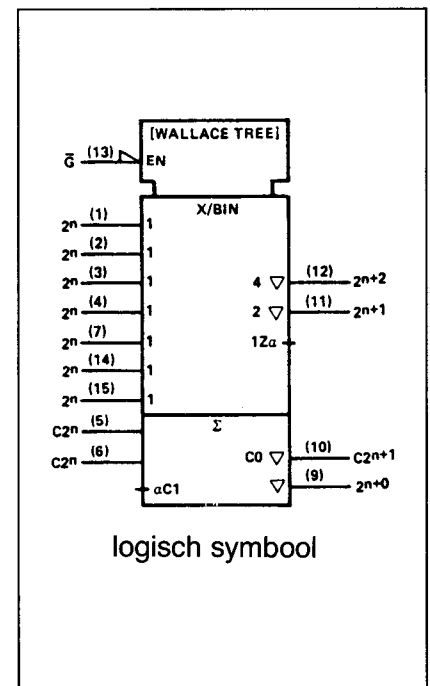
74275

7-bit-slice Wallace trees
met 3-state uitgangen

Figuur 6/4.2-275

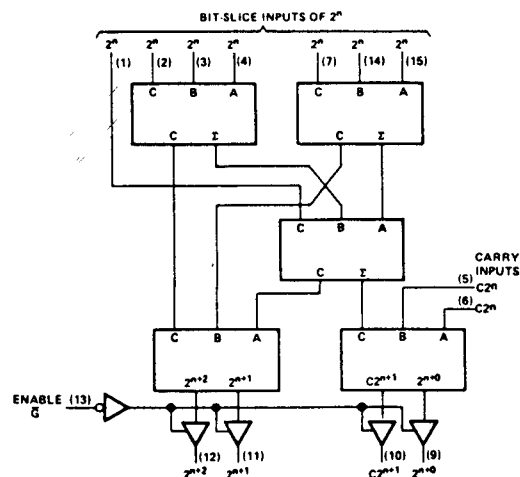


LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABLE PARAMETERS										Eenheid
I _{cc}				105	25					mA
I _{os}				-30 -100	-30 -130					mA
T _{plh} ¹⁾				50	35					ns
T _{p_{hl}} ¹⁾				50	42					ns
T _{pzh} ²⁾				15	8					ns
T _{p_{zl}} ²⁾				15	13					ns
T _{phz} ²⁾				10	10					ns
T _{p_{lz}} ²⁾				10	10					ns



¹⁾ Alle bit-slice of carry → alle uitgangen

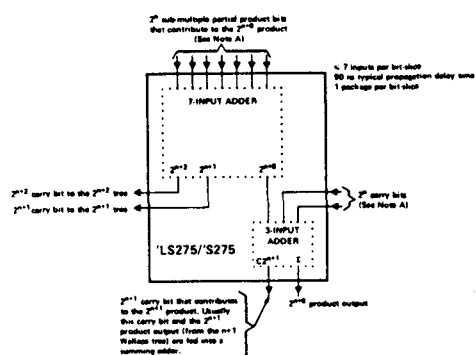
²⁾ \bar{G} → alle uitgangen



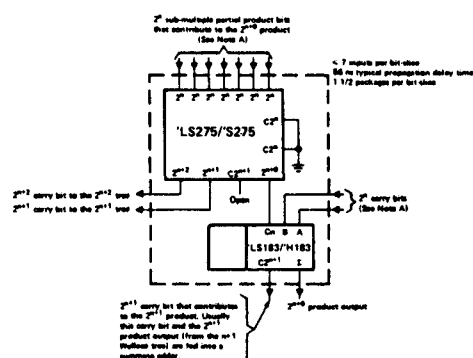
NOTE: When one of the C_2^n carry inputs is not used, it must be grounded. If neither C_2^n carry input is used, both C_2^n inputs are grounded and the C_2^{n+1} output is normally left open.

functioneel blokschema

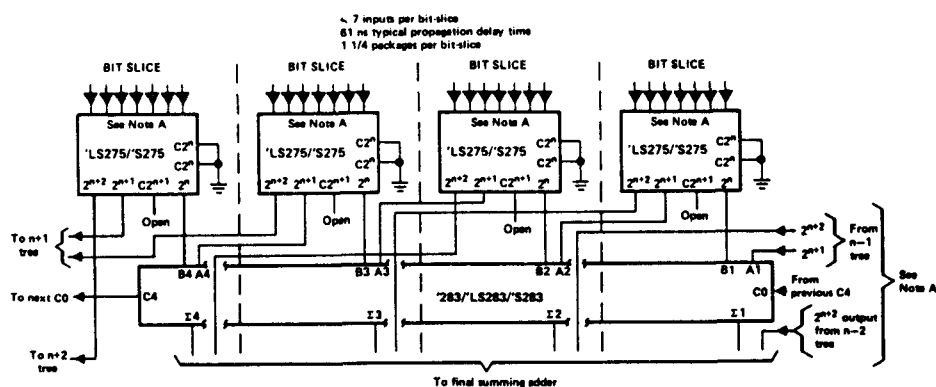
4.2 74xx-serie TTL en HC



-BASIC BIT-SLICE WALLACE TREE



-HIGH-SPEED BIT-SLICE WALLACE TREE

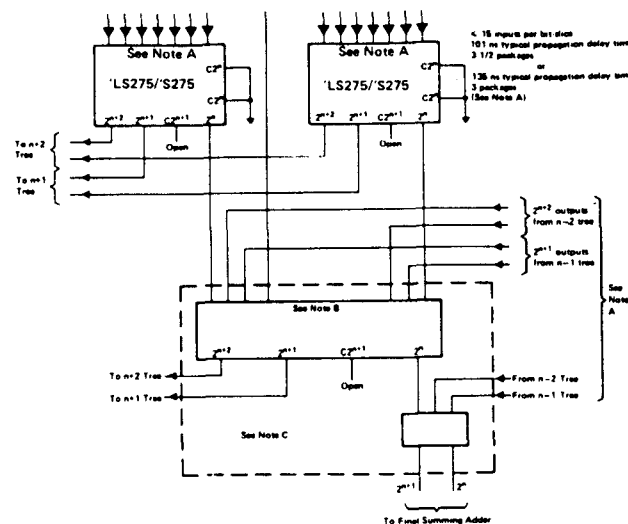


~~MODERATE-SPEED BIT-SLICE WALLACE TREE~~

NOTE A: All unused inputs must be grounded.

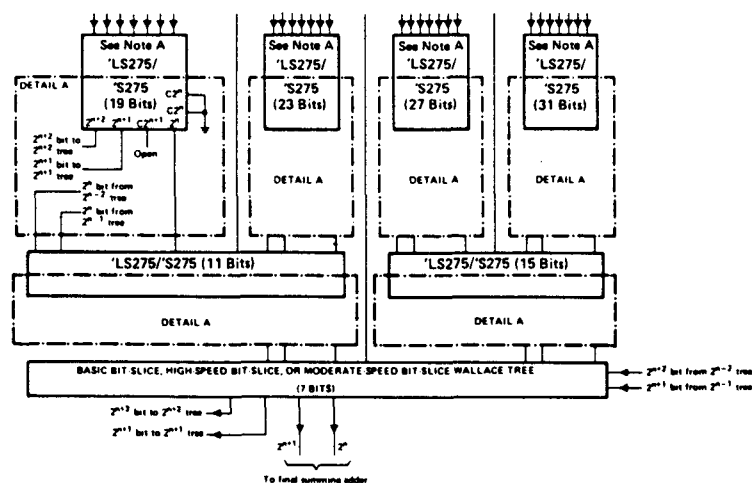
toepassingen

4.2 74xx-serie TTL en HC



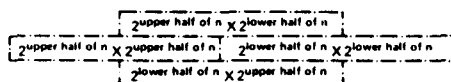
- NOTES: A. Ground unused inputs.
 B. These outputs from preceding trees may go to any of the inputs of the 'LS275/S275'.
 C. The circuit within the dotted lines may be either the basic bit-slice Wallace tree or the high-speed Wallace tree. In the latter case both carry inputs of the 'LS275/S275' must be grounded.

15-BIT-SLICE WALLACE TREE FOR 32-BIT X 32-BIT MULTIPLIER

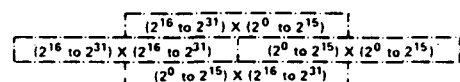


- NOTES: A. Ground unused inputs.
 B. The number of bits in parentheses is the maximum number of bits this tree can combine if the remaining 'LS275/S275' (all having a higher number in the parentheses) were not connected.

7-TO-31-BIT-SLICE WALLACE TREE FOR UP TO 64-BIT X 64-BIT MULTIPLIERS



NOTE A: The left-hand half of each rectangle is the portion of word one used to obtain the product shown within the rectangle. Similarly, the right-hand half of each rectangle is the portion of word two used.

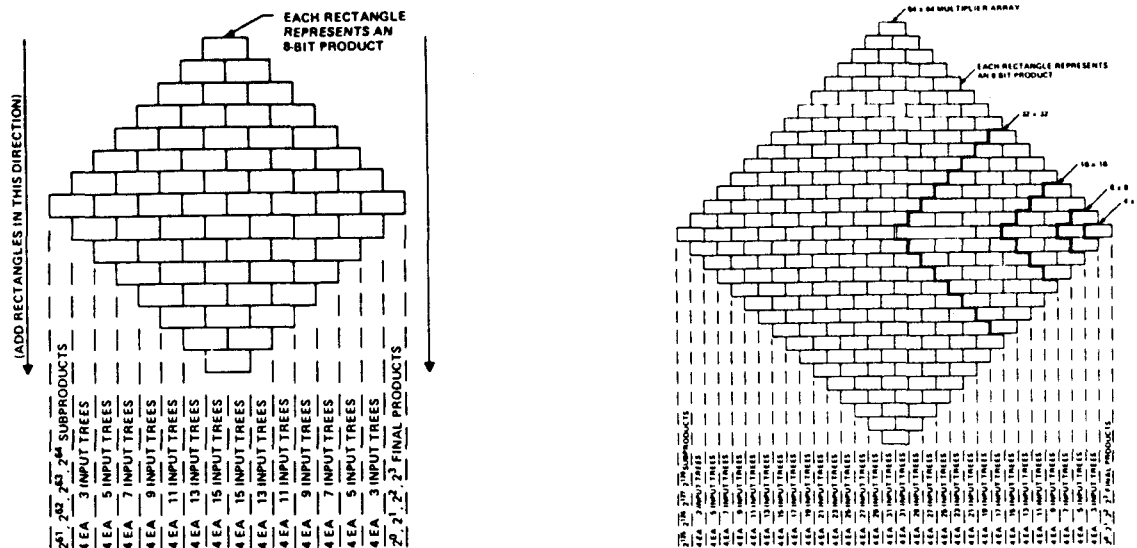


Methode om 32-bit producten op te tellen voor het verkrijgen van een 64-bit produkt.

Universele methode voor het optellen van $\frac{n}{2}$ -bit producten om een n-bit produkt te verkrijgen.

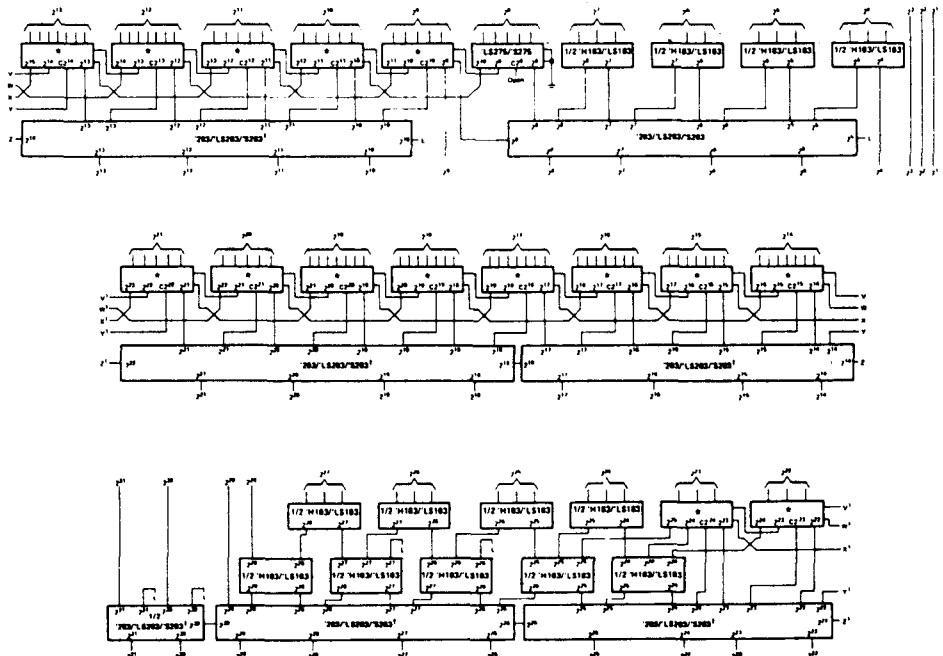
Deel 6: Digitale rekenkundige schakelingen

4.2 74xx-serie TTL en HC



Eindprodukten en array subprodukt optellingen voor een 32-bit x 32-bit vermenigvuldiger.

Array schikking voor verschillende vermenigvuldigers, inclusief array subprodukt optellingen voor 64-bit x 64-bit multiplier.



figuur C
(zie ook figuren
A en B bij 27274)

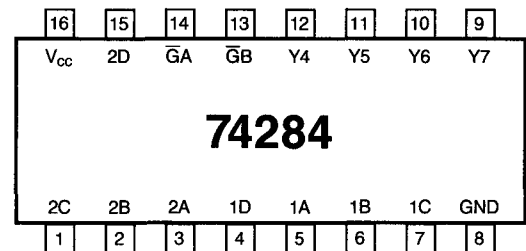
*Each starred block may be either a basic bit-slice Wallace tree ('LS275 or 'S275 only) or a high-speed bit-slice Wallace tree ('LS275 plus 1/2 'LS183 or 'S275 plus 1/2 'H183). In either case the function of the terminal is the same as the similarly located terminal of the basic bit-slice (Figure 1) or high-speed bit-slice Wallace tree (Figure 2). Also for either tree, when only five inputs of the seven-input adder of the 'LS275/'S275 are used, the remaining two inputs must be grounded. When the high-speed adder is used, the C2⁰ inputs of the 'LS275/'S275 must be grounded.

†For improved performance SN74LS181/SN74S181 ALUs with SN74S182 look-ahead generators can be substituted for the SN74283/SN74LS283/SN74S283 adders. Typically, the multiplication time will be reduced by 18 to 32 nanoseconds.

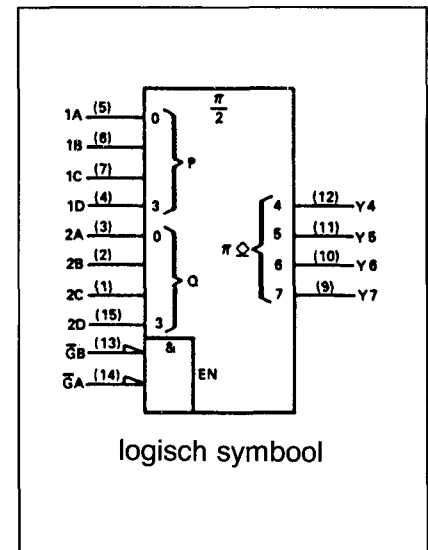
4.2 74xx-serie TTL en HC

74284**4-bit by 4-bit parallelle binaire multiplier**
(gebruiken met 74285)

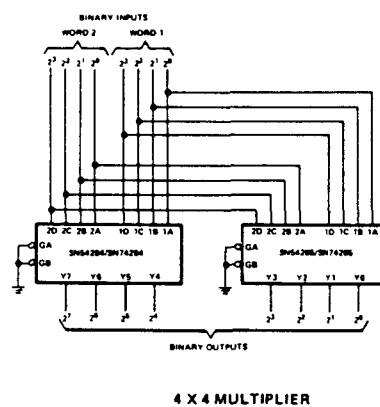
Figuur 6/4.2-284



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}	92									mA
I _{ol}	16									mA
T _{plh} ¹⁾	20									ns
T _{phl} ¹⁾	20									ns
T _{plh} ²⁾	40									ns
T _{phl} ²⁾	40									ns



1) \overline{G}_A of $\overline{G}_B \rightarrow Y$
 2) A,B,C,D $\rightarrow Y$



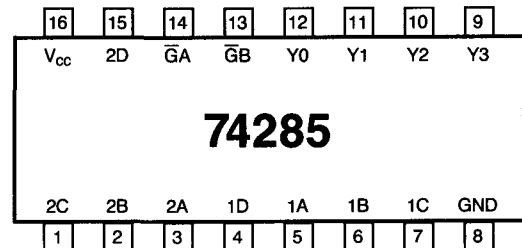
toepassing
(samen met 74285)

4.2 74xx-serie TTL en HC

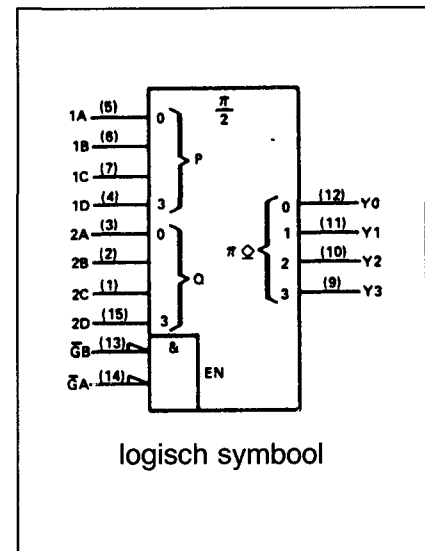
74285

4-bit by 4-bit parallelle binaire
multiplier
(gebruiken met 74284)

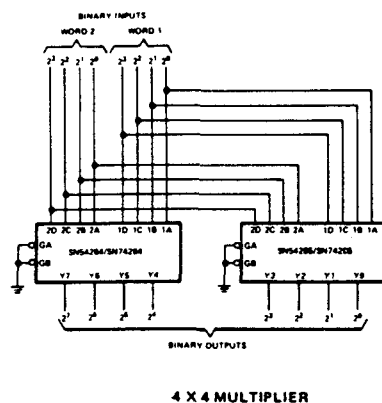
Figuur 6/4.2-285



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}	92									mA
I _{ol}	16									mA
T _{plh} ¹⁾	20									ns
T _{phl} ¹⁾	20									ns
T _{plh} ²⁾	40									ns
T _{phl} ²⁾	40									ns



- 1) $\overline{G_A}$ of $\overline{G_B} \rightarrow Y$
 2) A, B, C of D $\rightarrow Y$



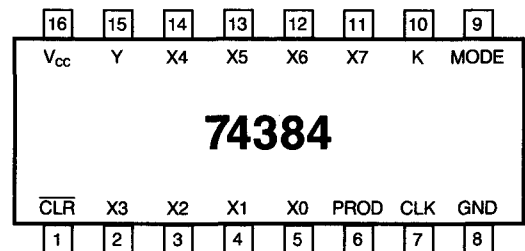
toepassing
(samen met 74284)

4.2 74xx-serie TTL en HC

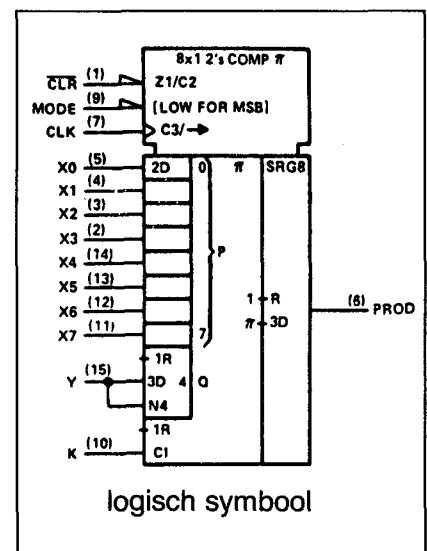
74384

8-bit by 1-bit two's-complement
multiplier
(gelijk aan 25LS14)

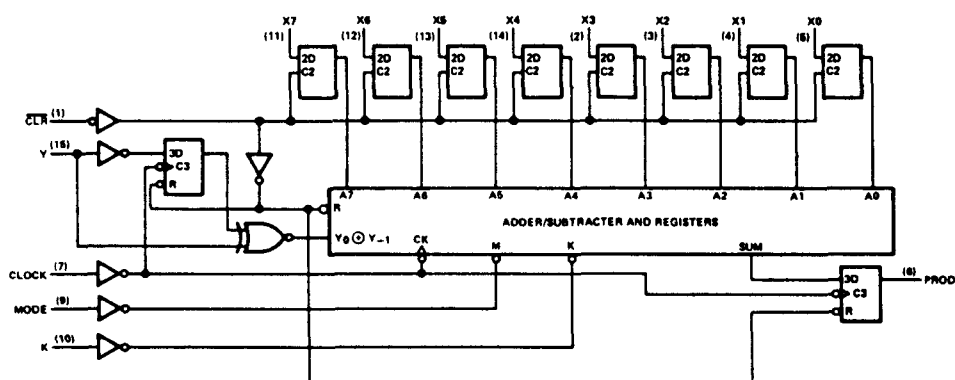
Figuur 6/4.2-384



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}			60		91					mA
I _{os}			-60 -150		-20 -100					mA
T _{plh} ¹⁾			6.5		15					ns
T _{phl} ¹⁾			6.5		15					ns
T _{phl} ²⁾			10		17					ns
f _{max}			100		40					MHz



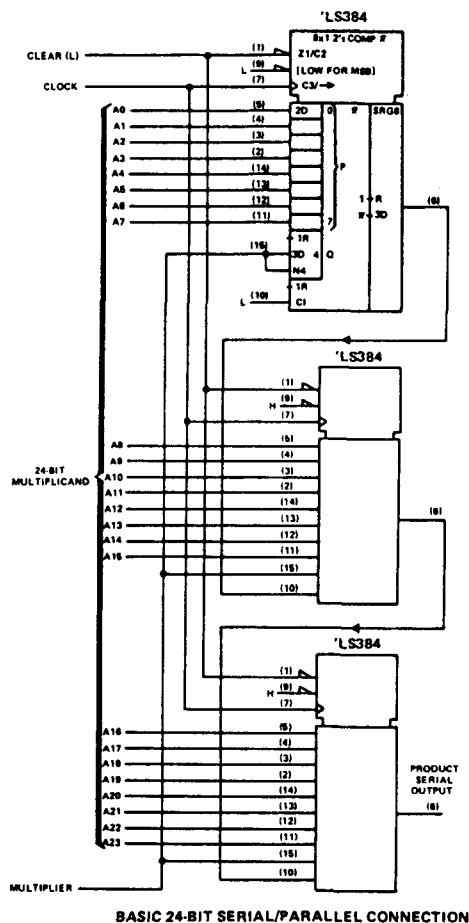
- 1) Clock → uitgang
2) Clear → uitgang



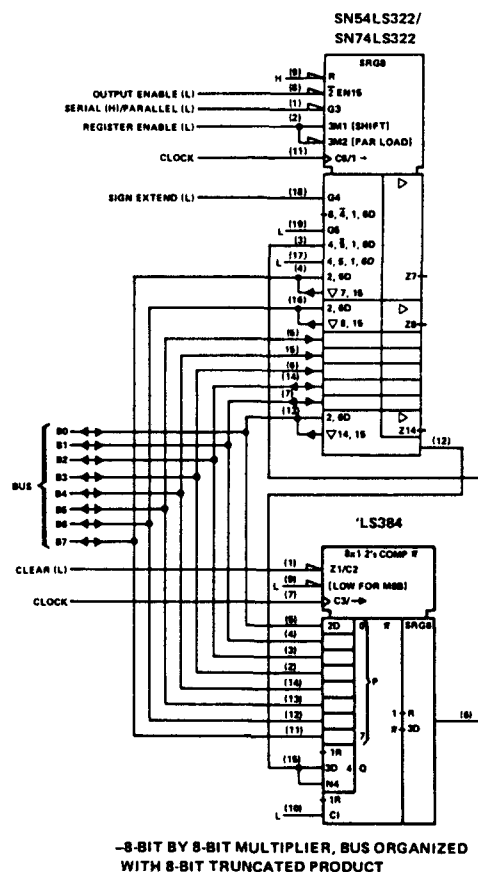
funtioneel blokschema (pos. logika)

4.2 74xx-serie TTL en HC

toepassingen



BASIC 24-BIT SERIAL/PARALLEL CONNECTION



-8-BIT BY 8-BIT MULTIPLIER, BUS ORGANIZED WITH 8-BIT TRUNCATED PRODUCT

6/4.3

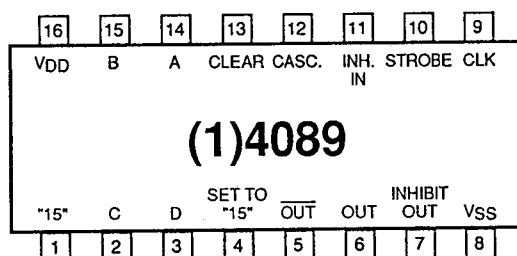
Binaire multipliers
(1)4xxx-serie CMOS

(1)4089

Binaire rate multiplier

- voeding 3 tot 15 V
- interne synchrone 4-bit teller
- STROBE, INHIBIT en CASCADE-ingangen
- CLEAR en SET-ingangen
- complementaire uitgangen
- "15" en INHIBIT uitgang
- $f_{out} = M \cdot f_{in} / 16$
($M = D \cdot 2^3 + C \cdot 2^2 + B \cdot 2^1 + A \cdot 2^0$)

Figuur 6/4.3-89

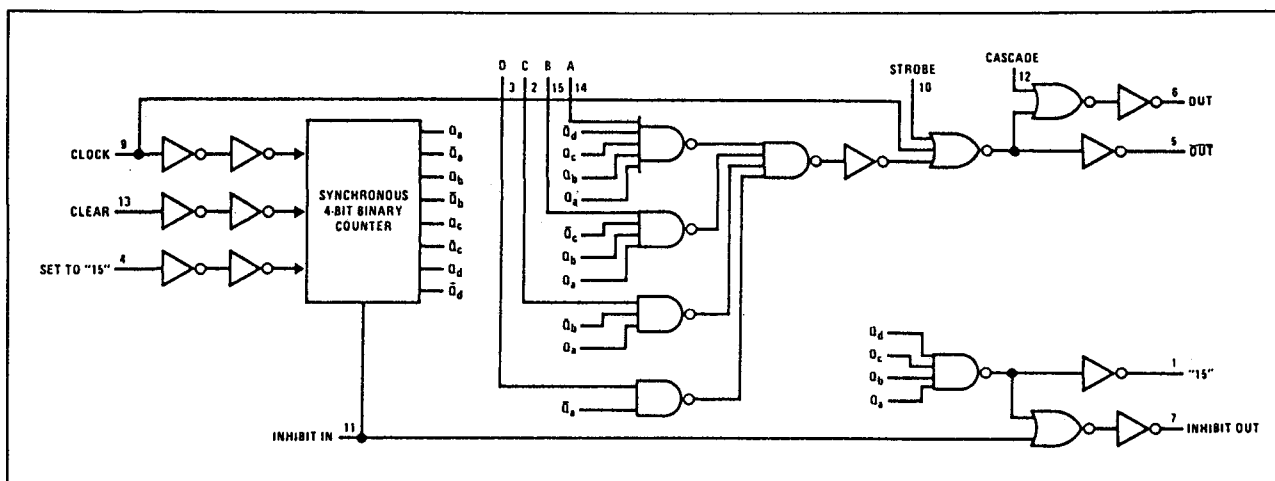


INPUTS										NUMBER OF PULSES OR OUTPUT LOGIC LEVEL (H OR L)			
D	C	B	A	No. of Clock Pulses	Inh In	Strobe	Cascade	Clear	Set	Pin 6 Out	Pin 5 Out	Pin 7 Inh Out	Pin 1 "15"
0	0	0	0	16	0	0	0	0	0	L	H	1	1
0	0	0	1	16	0	0	0	0	0	1	1	1	1
0	0	1	0	16	0	0	0	0	0	2	2	1	1
0	0	1	1	16	0	0	0	0	0	3	3	1	1
0	1	0	0	16	0	0	0	0	0	4	4	1	1
0	1	0	1	16	0	0	0	0	0	5	5	1	1
0	1	1	0	16	0	0	0	0	0	6	6	1	1
0	1	1	1	16	0	0	0	0	0	7	7	1	1
1	0	0	0	16	0	0	0	0	0	8	8	1	1
1	0	0	1	16	0	0	0	0	0	9	9	1	1
1	0	1	0	16	0	0	0	0	0	10	10	1	1
1	0	1	1	16	0	0	0	0	0	11	11	1	1
1	1	0	0	16	0	0	0	0	0	12	12	1	1
1	1	0	1	16	0	0	0	0	0	13	13	1	1
1	1	1	0	16	0	0	0	0	0	14	14	1	1
1	1	1	1	16	0	0	0	0	0	15	15	1	1
X	X	X	X	16	1	0	0	0	0	Depends on internal state of counter L H * 1 1			
X	X	X	X	16	0	1	0	0	0				
X	X	X	X	16	0	0	1	0	0				
1	X	X	X	16	0	0	0	1	0	16	16	H	L
0	X	X	X	16	0	0	0	1	0	L	H	H	L
X	X	X	X	16	0	0	0	0	1	L	H	L	H

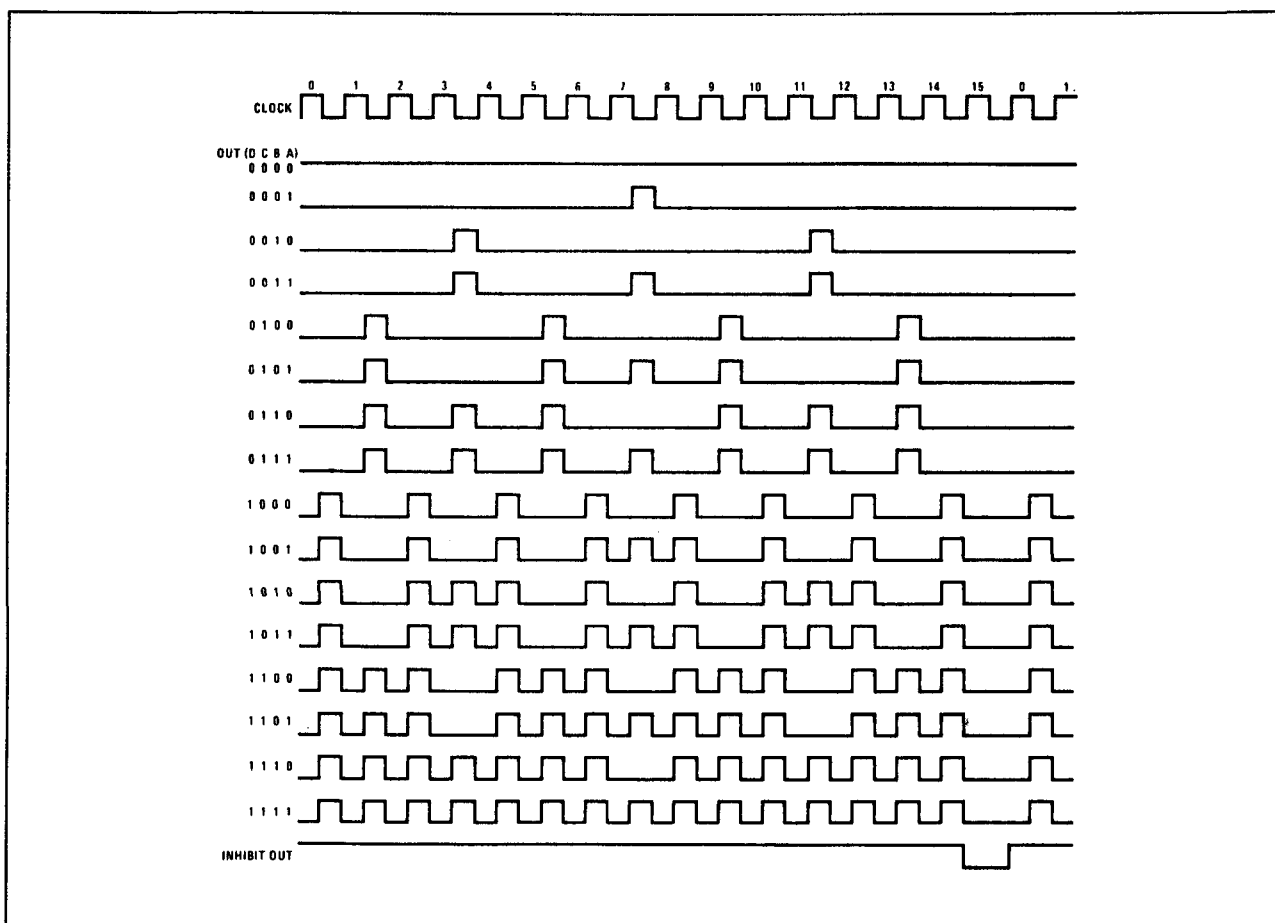
*Output same as the first 16 lines of this truth table (depending on values of A, B, C, D)

Waarheidstabel

4.3 Binaire multipliers (1)4xxx-serie CMOS

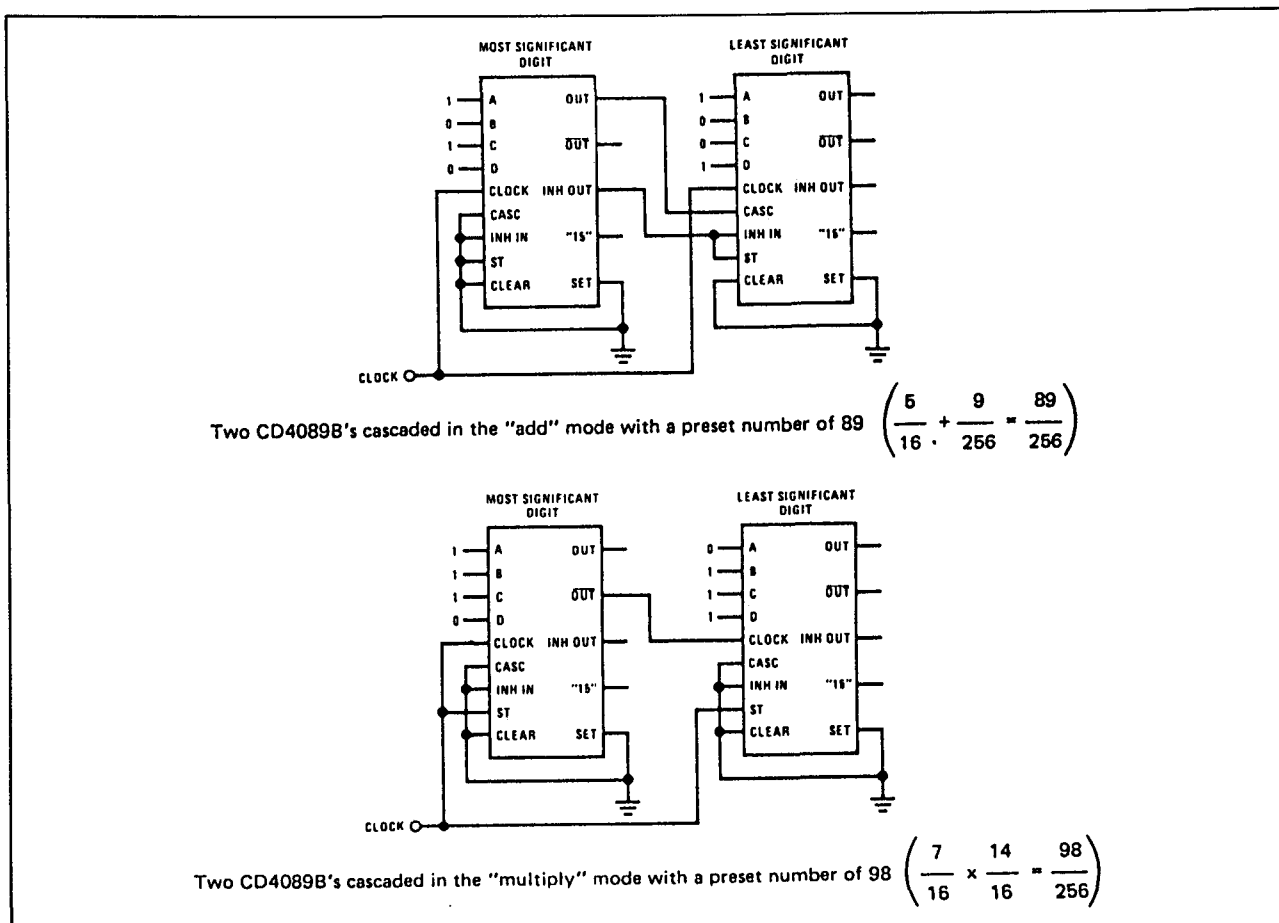


Functioneel blokschema, positieve logica



Logische golfvormen

4.3 Binaire multipliers (1)4xxx-serie CMOS



Cascadeschakeling van twee IC's

o.a. leverbaar:
 CD 4089B, HCC 4089B

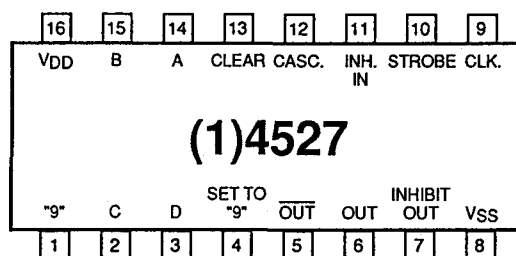
4.3 Binaire multipliers (1)4xxx-serie CMOS

(1)4527

BCD rate multiplier

- voeding 3 tot 15 V
- interne synchrone 4-bit teller
- STROBE, INHIBIT en CASCADE-ingangen
- CLEAR en SET-ingangen
- complementaire uitgangen
- "9" en INHIBIT uitgang
- $f_{out} = M \cdot f_{in} / 10$
 $(M = D \cdot 2^3 + C \cdot 2^2 + B \cdot 2^1 + A \cdot 2^0)$

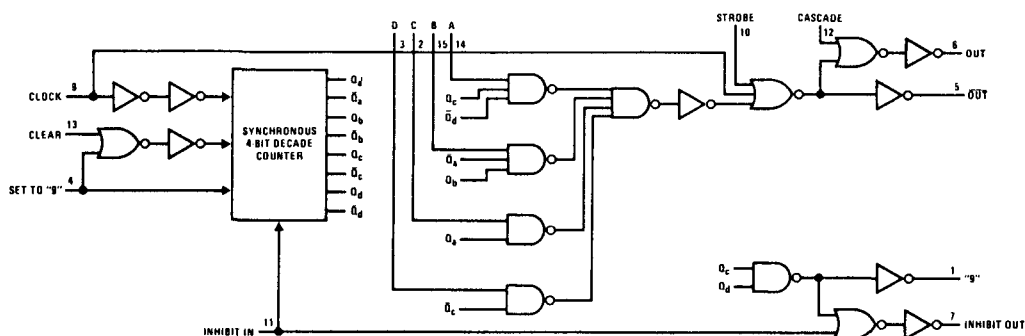
Figuur 6/4.3-527



INPUTS										NUMBER OF PULSES OR OUTPUT LOGIC LEVEL (H OR L)			
D	C	B	A	No. of Clock Pulses	Inh _{IN}	Strobe	Cascade	Clear	Set	Pin 6 Out	Pin 5 Out	Pin 7 Inh Out	Pin 1 "9"
0	0	0	0	10	0	0	0	0	0	L	H	1	1
0	0	0	1	10	0	0	0	0	0	0	1	1	1
0	0	1	0	10	0	0	0	0	0	0	2	2	1
0	0	1	1	10	0	0	0	0	0	0	3	3	1
0	1	0	0	10	0	0	0	0	0	0	4	4	1
0	1	0	1	10	0	0	0	0	0	0	5	5	1
0	1	1	0	10	0	0	0	0	0	0	6	6	1
0	1	1	1	10	0	0	0	0	0	0	7	7	1
1	0	0	0	10	0	0	0	0	0	0	8	8	1
1	0	0	1	10	0	0	0	0	0	0	9	9	1
1	0	1	0	10	0	0	0	0	0	0	8	8	1
1	0	1	1	10	0	0	0	0	0	0	9	9	1
1	1	0	0	10	0	0	0	0	0	0	8	8	1
1	1	0	1	10	0	0	0	0	0	0	9	9	1
1	1	1	0	10	0	0	0	0	0	0	8	8	1
1	1	1	1	10	0	0	0	0	0	0	9	9	1
X	X	X	X	10	1	0	0	0	0	Depends on internal state of counter			
X	X	X	X	10	0	1	0	0	0				
X	X	X	X	10	0	0	1	0	0	H	H	1	1
1	X	X	X	10	0	0	0	1	0	10	10	H	L
0	X	X	X	10	0	0	0	1	0	L	H	H	L
X	X	X	X	10	0	0	0	0	1	L	H	L	H

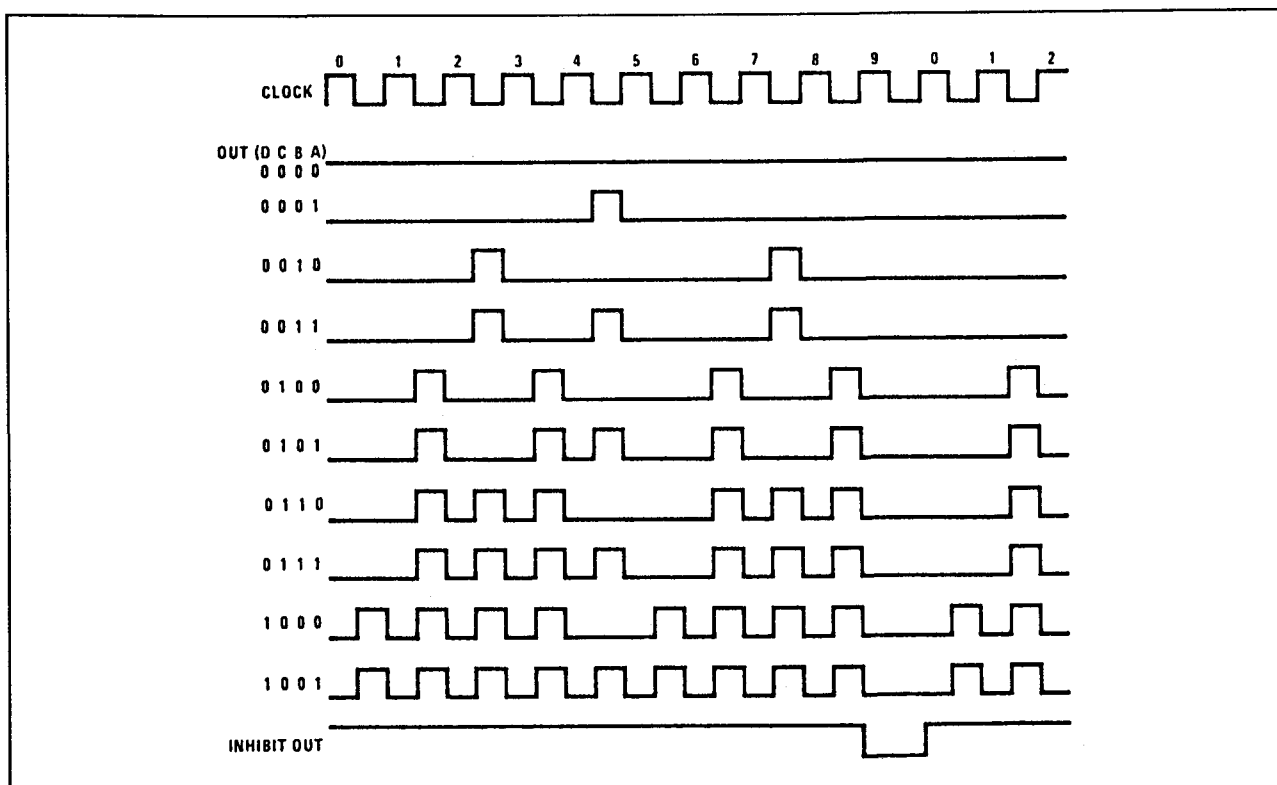
* Output same as the first 16 lines of this truth table (depending on values of A, B, C, D)

Waarheidstabel

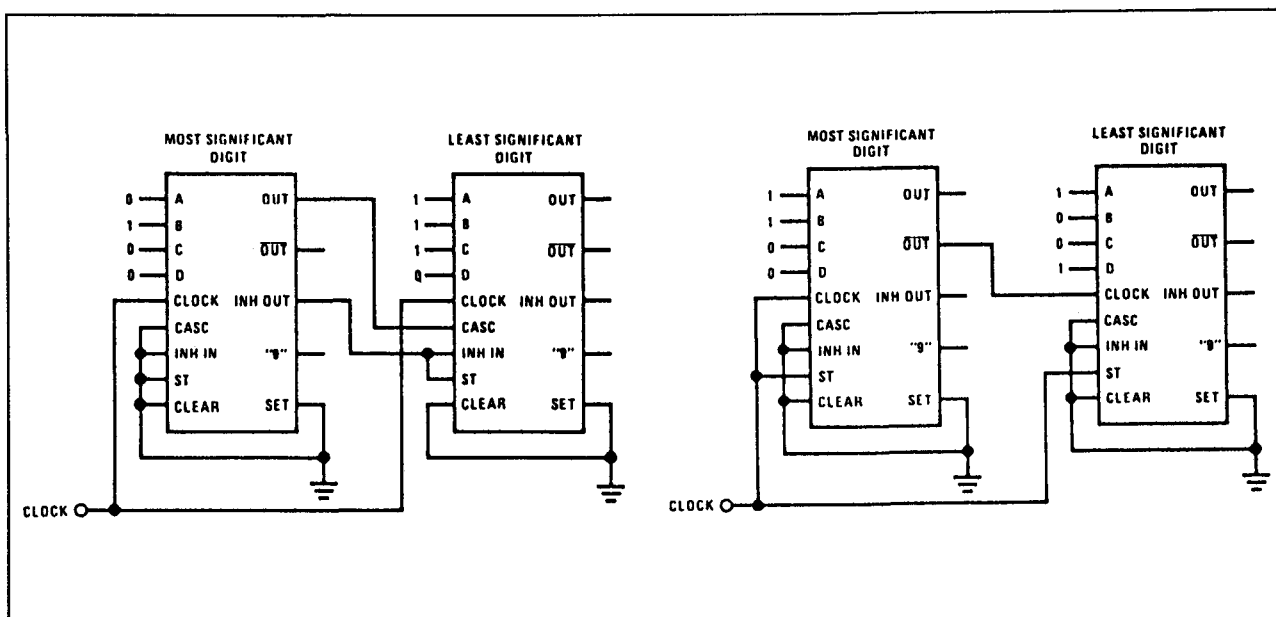


Functioneel blokschema, positieve logica

4.3 Binaire multipliers (1)4xxx-serie CMOS



Logische golfvormen



Cascadeschakelingen van twee schakelingen

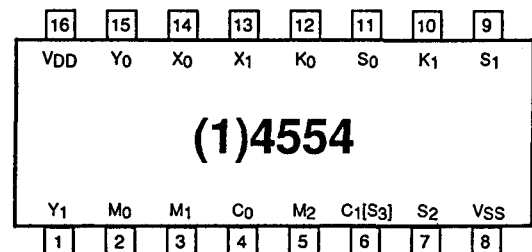
o.a. leverbaar:
CD 4527B, MC 14527B, HEF 4527 B

4.3 Binaire multipliers (1)4xxx-serie CMOS

(1)4554**2-bit x 2-bit binaire multiplier**

- voeding 3 tot 18 V
- gelijktijdig vermenigvuldigen en optellen
- gemakkelijke m-bit x n-bit uitbreiding
- bij uitbreiding GEEN extra logika nodig

Figuur 6/4.3-554

**Ingangen**

X0 en X12 (vermenigvuldigtal),
Y0 en Y1 (vermenigvuldiger),
K0 en K1 (cascade)
M0, M1 en M2 (optel)

Uitgangen

S0, S1 en S2 (som)
C1 [S3] en C0 (carry)

$$S = (X \times Y) + K + M$$

waarin (binaire getallen):

$$S = S3 S2 S1 S0$$

$$X = X1 X0$$

$$Y = Y1 Y0$$

$$K = K1 K0$$

$$M = M1 M0$$

Voorbeeld

als:

$$X = 2 \text{ (10)}$$

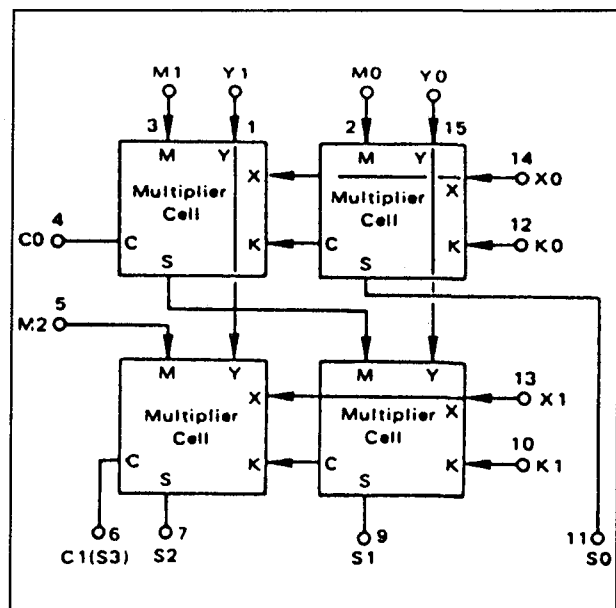
$$Y = 3 \text{ (11)}$$

$$K = 1 \text{ (01)}$$

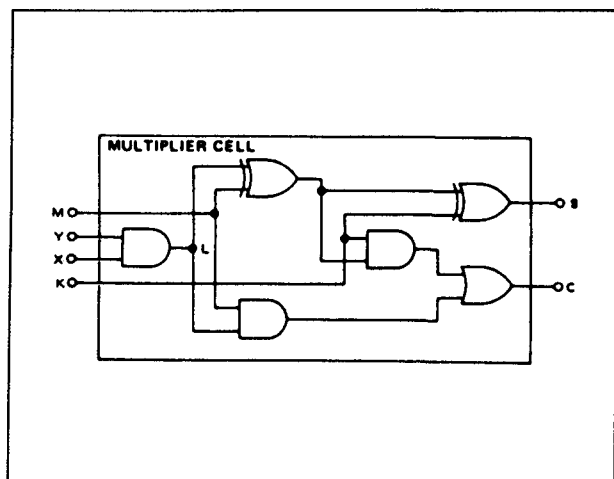
$$M = 2 \text{ (10)}$$

dan is:

$$S = (2 \times 3) + 1 + 2 = 9$$

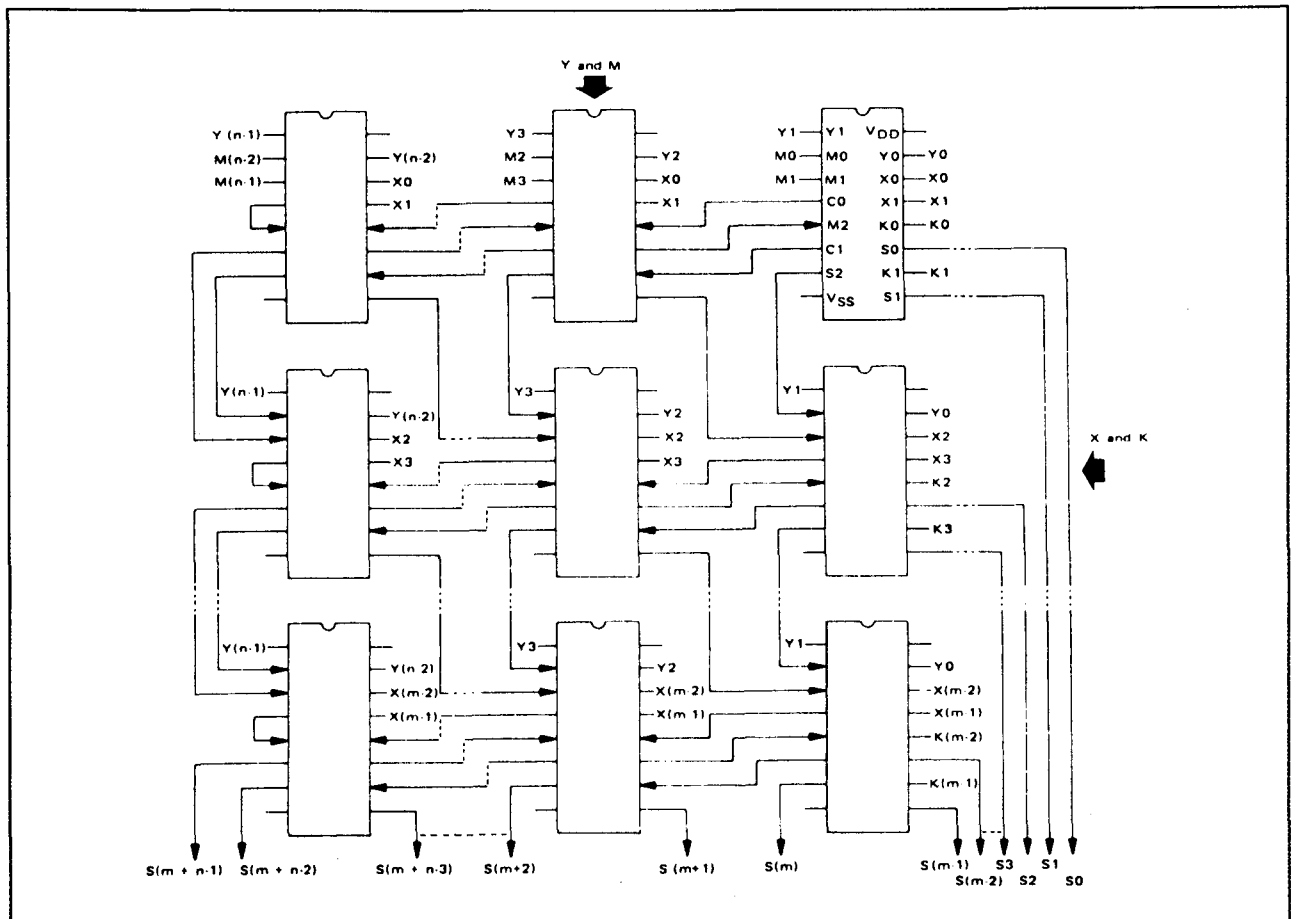


Functioneel blokschema, positieve logica



Opbouw van één vermenigvuldigcel

4.3 Binaire multipliers (1)4xxx-serie CMOS

Cascadeschakeling voor m -bit \times n -bit

o.a. leverbaar:
 HD 14554 B, MC 14554 B

4.3 Binaire multipliers (1)4xxx-serie CMOS

6/5

Pariteitsgeneratoren

Inhoud

6/5 Pariteitsgeneratoren 74xx-serie TTL en HC

(aanvulling 17)

74180 9-bit pariteitsgenerator/tester

74280 9-bit pariteitsgenerator/tester

74286 9-bit pariteitsgenerator/tester, par. I/O-busdriver

6/5.2

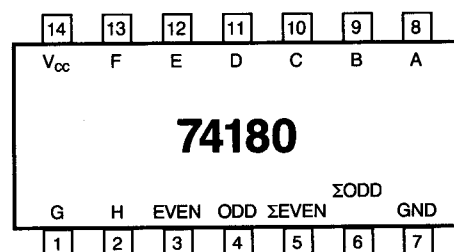
Pariteitsgeneratoren 74xx-serie TTL en HC

74180

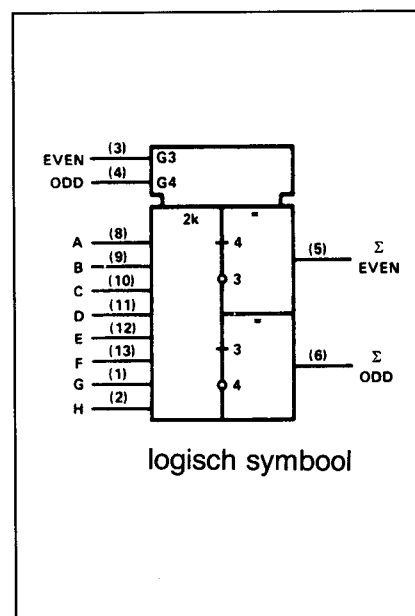
9-bit even/oneven pariteits-
generator/tester

(8-bit data + 1 pariteitsbit)

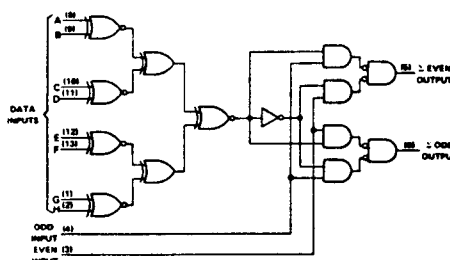
Figuur 6/5.2-180



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}	34								80 ⁶⁾	mA
I _{os}	-18 -55								4	mA
T _{plh} / T _{phl} ¹⁾	40/ 45								36	ns
T _{plh} / T _{phl} ²⁾	32/ 25								33	ns
T _{plh} / T _{phl} ³⁾	32/ 25								36	ns
T _{plh} / T _{phl} ⁴⁾	40/ 45								33	ns
T _{plh} / T _{phl} ⁵⁾	13/ 7								15	ns



- 1) data → Σ even
- 2) data → Σ odd (oneven)
- 3) data → Σ even
- 4) data → Σ odd (oneven)
- 5) even of odd → Σ even of Σ odd
- 6) μA



functioneel blokschema

FUNCTION TABLE				
INPUTS			OUTPUTS	
Σ OF H's AT A THRU H	EVEN	ODD	Σ EVEN	Σ ODD
EVEN	H	L	H	L
ODD	H	L	L	H
EVEN	L	H	L	H
ODD	L	H	H	L
X	H	H	L	L
X	L	L	H	H

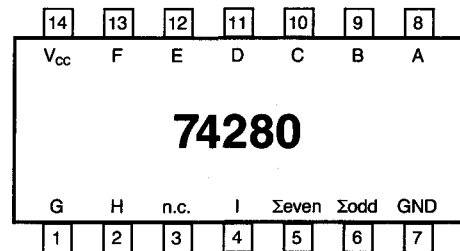
H = high level, L = low level, X = irrelevant

waarheidstabel

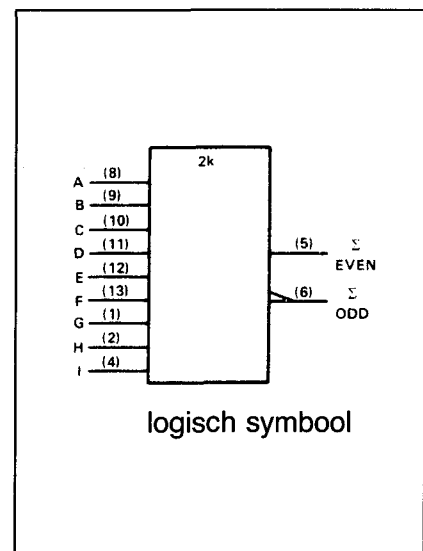
5.2 74xx-serie TTL en HC

74280**9-bit even/oneven pariteits-
generator/tester**
(8-bit data + 1 pariteitsbit)

Figuur 6/5.2-280



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Een- heid
I _{cc}			25	67 105	16	25			80 ³⁾	mA
I _{os}			-60 -150	-40 -100	-20 -100	-30 -112			4	mA
T _{plh} ¹⁾			10	14	33	2 14			21	ns
T _{p_{hl}} ¹⁾			11	11,5	29	2 14			21	ns
T _{plh} ²⁾			10	14	23	2 14			21	ns
T _{p_{hl}} ²⁾			11	11,5	31	2 14			21	ns

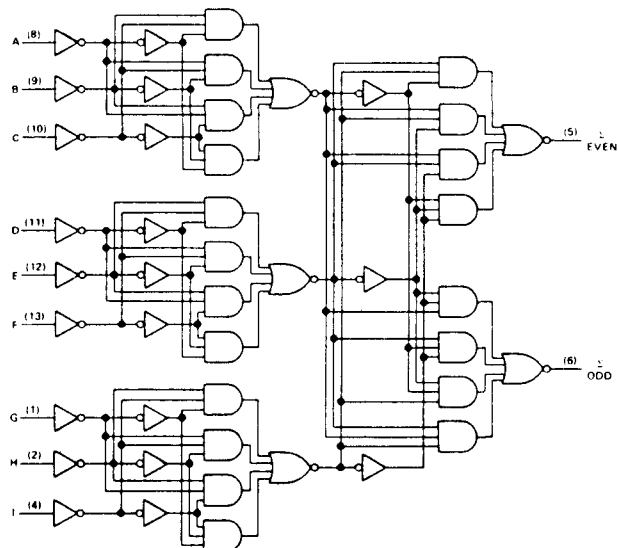


- 1) data → Σ even
 2) data → Σ odd (oneven)
 3) μA

FUNCTION TABLE		
NUMBER OF INPUTS A THRU I THAT ARE HIGH	OUTPUTS	
	Σ EVEN	Σ ODD
0, 2, 4, 6, 8	H	L
1, 3, 5, 7, 9	L	H

H = high level, L = low level

waarheidstabel



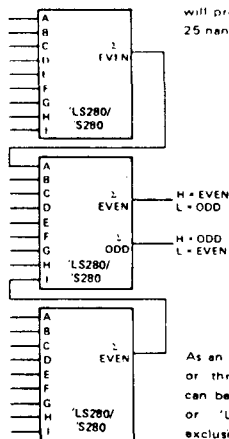
functioneel blokschema

5.2 74xx-serie TTL en HC

TYPICAL APPLICATION DATA

25-LINE PARITY/GENERATOR CHECKER

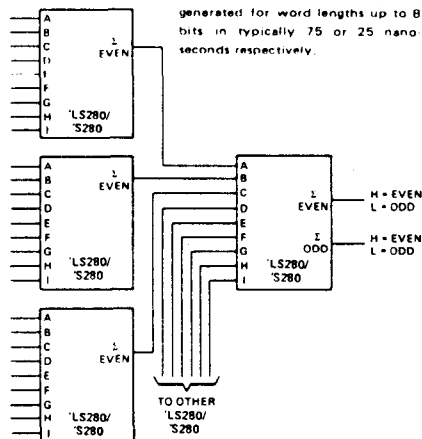
Three 'LS280's or 'S280's can be used to implement a 25 line parity generator/checker. This arrangement will provide parity in typically 75 or 25 nanoseconds respectively.



As an alternative, the outputs of two or three parity generators/checkers can be decoded with a 2-input ('S86 or 'LS86) or 3-input ('S135) exclusive-OR gate for 18- or 27-line parity applications.

81-LINE PARITY/GENERATOR CHECKER

Longer word lengths can be implemented by cascading 'LS280's or 'S280's. As shown here, parity can be generated for word lengths up to 81 bits in typically 75 or 25 nanoseconds respectively.



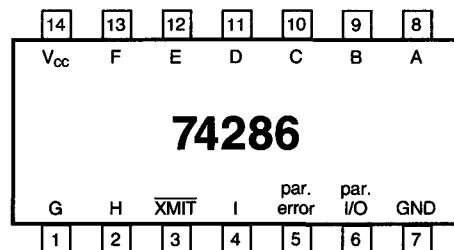
enkele toepassingen

5.2 74xx-serie TTL en HC

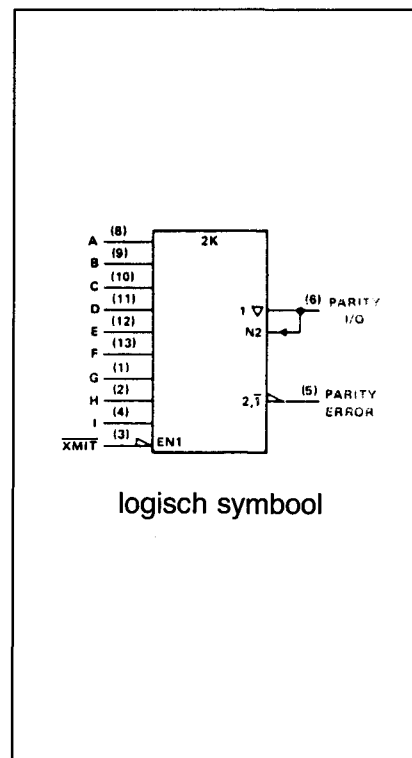
74286

9-bit pariteitsgenerator/tester
met bus-driver (pariteits in-/uitgangspoort)

Figuur 6/5.2-286



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
$I_{CC} \begin{smallmatrix} T \\ R \end{smallmatrix}$						30 35				mA
I_{OS}						-30 -112				mA
$T_{plh}/T_{phl} \begin{smallmatrix} 1) \\ \end{smallmatrix}$						3 15				ns
$T_{plh}/T_{phl} \begin{smallmatrix} 2) \\ \end{smallmatrix}$						3 16,5				ns
$T_{plh}/T_{phl} \begin{smallmatrix} 3) \\ \end{smallmatrix}$						3 9				ns
$T_{pzh} \begin{smallmatrix} 4) \\ \end{smallmatrix}$						3 13				ns
$T_{pzi} \begin{smallmatrix} 4) \\ \end{smallmatrix}$						3 16				ns
$T_{phz} \begin{smallmatrix} 5) \\ \end{smallmatrix}$						3 11,5				ns
$T_{plz} \begin{smallmatrix} 5) \\ \end{smallmatrix}$						3 10				ns



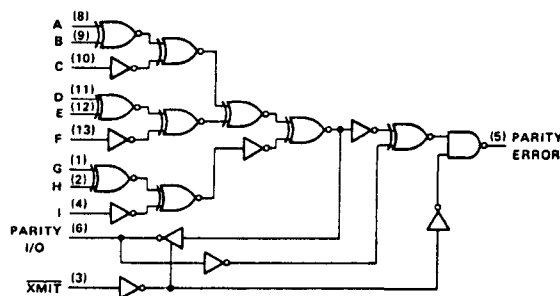
- 1) A t/m I → parity I/O
 2) A t/m I → parity error
 3) parity I/O → parity error
 4) XMIT → parity I/O (enable)
 5) idem (disable)

FUNCTION TABLE

NUMBER OF INPUTS (A THRU I) THAT ARE HIGH	\overline{XMIT}	PARITY I/O	PARITY ERROR
0, 2, 4, 6, 8	L	H	H
1, 3, 5, 7, 9	L	L	H
0, 2, 4, 6, 8	H	H	H
	H	L	L
1, 3, 5, 7, 9	H	H	L
	H	L	H

h - high input level
 H - high output level
 l - low input level
 L - low output level

waarheidstabel



functioneel blokschema

5.2 74xx-serie TTL en HC

TYPICAL APPLICATION DATA

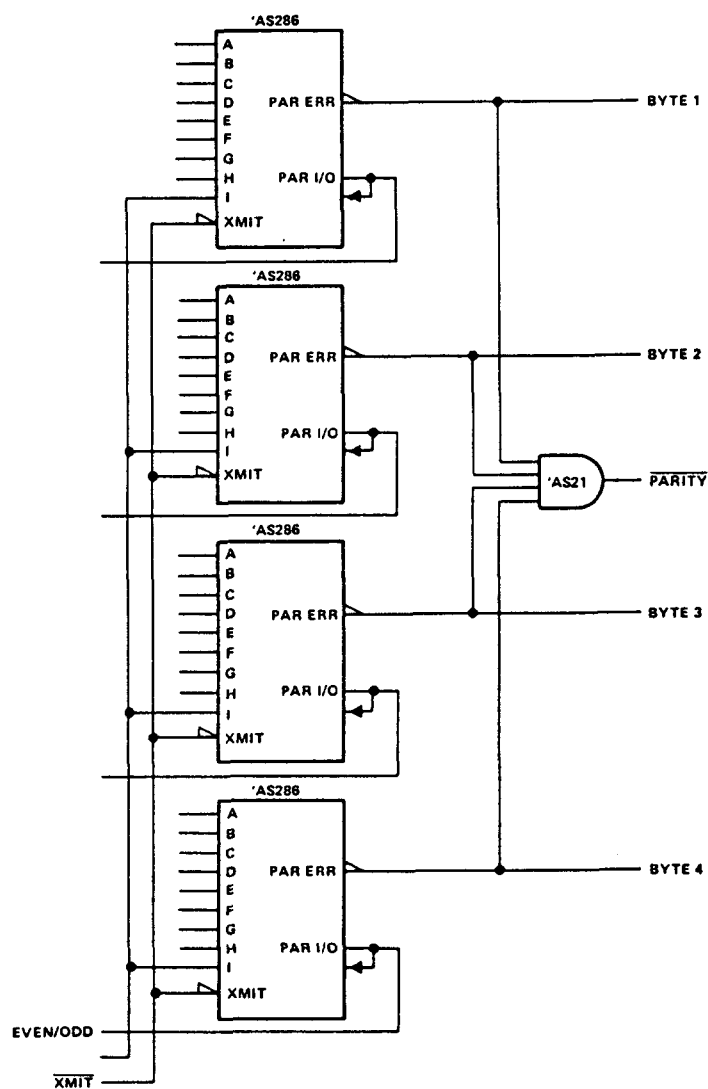


FIGURE 1—32-BIT PARITY GENERATOR/CHECKER

32-bit parity generator/checker with output polarity-switching, parity error detection, and parity on every byte.

toepassing

5.2 74xx-serie TTL en HC

6/6

Arithmetic Logic Units, Look-ahead carry-generatoren en binaire accumulators

Inhoud

6/6.2 ALU's, carry-generatoren en binaire accumulators 74xx-serie TTL en HC

(aanvulling 17)

74181	ALU/functiegenerator
74182	carry-generator
74281	binaire accumulator
74282	carry-generator, selecteerbare carry-ingangen
74381	ALU/functiegenerator
74382	ALU/functiegenerator
74681	binaire accumulator
74881	ALU/functiegenerator
74882	carry-generator, 32-bit

6/6.2

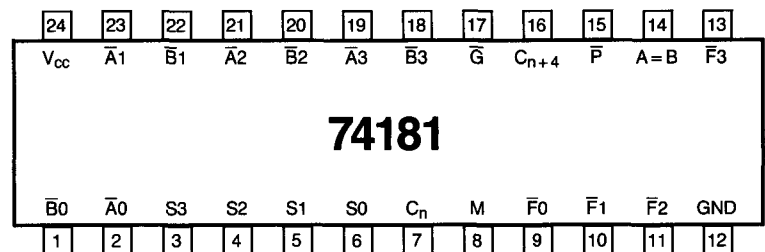
ALU's, carry-generatoren en binaire accumulatoren 74xx-serie TTL en HC

74181

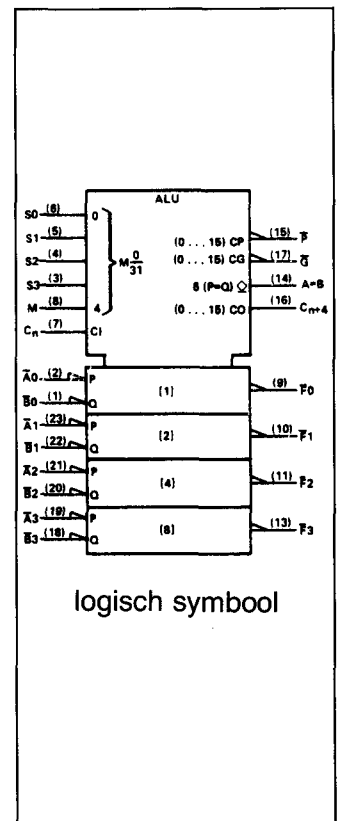
Arithmetic Logic Unit

ALU/functie generator

Figuur 6/6.2-181



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
Icc	94		43	120	21	135			80 ⁹⁾	mA
Ios	-18 -57		-60 -150	-40 -100	-5 -42	-30 -112			4	mA
Tplh/ Tphl ¹⁾	12/ 13		6.4/ 6.1	7/ 7	18/ 13	5			13	ns
Tplh/ Tphl ²⁾	28/27 35/33		10/9.4 10.8/10	12.5/12.5 15.5/15.5	25/25 27/27	6 7			30 30	ns
Tplh/ Tphl ³⁾	13/ 12		6.7/ 6.5	7/ 7	17/ 13	5			20	ns
Tplh/ Tphl ⁴⁾	13/13 17/17		5.7/5.8 6.5/7.3	8/7.5 10.5/10.5	19/15 21/21	4 5			20 20	ns
Tplh/ Tphl ⁵⁾	13/17 17/17		5/5.5 5.8/6.5	7.5/7.5 10.5/10.5	20/20 20/22	5 5			27 24	ns
Tplh/ Tphl ⁶⁾	28/21 32/23		7/7.2 8.2/5	11/11 14/14	21/13 21/21	5 5			20 19	ns
Tplh/ Tphl ⁷⁾	32/ 23		6/ 6	14/ 14	22/ 26	6			25	ns
Tplh/ Tphl ⁸⁾	35/ 32		18.5/ 9.8	15/ 20	33/ 41	12			25	ns



- ¹⁾ $C_n \rightarrow C_{n+4}$ ⁴⁾ \bar{A} of $\bar{B} \rightarrow \bar{G}$ { SUM mode, DIFF mode } ⁶⁾ \bar{A}_i of $\bar{B}_i \rightarrow \bar{F}_i$ { SUM mode, DIFF mode } ⁷⁾ \bar{A}_i of $\bar{B}_i \rightarrow \bar{F}_i$ (logic mode)
²⁾ \bar{A} of $\bar{B} \rightarrow C_{n+4}$ { SUM mode, DIFF mode } ⁵⁾ \bar{A} of $\bar{B} \rightarrow \bar{P}$ { SUM mode, DIFF mode } ⁸⁾ \bar{A} of $\bar{B} \rightarrow A=B$ (DIFF mode)
³⁾ $C_n \rightarrow \bar{F}$ ⁹⁾ μA

Met de ALU kunnen 16 binaire rekenkundige bewerkingen worden uitgevoerd op twee 4-bit woorden (zie tabellen 1 en 2).

Wanneer 'Look-ahead carry generators' (74182) worden gebruikt, kunnen ook grotere woorden snel worden verwerkt (tabel 'typical addition times')

NUMBER OF BITS	TYPICAL ADDITION TIMES			PACKAGE COUNT		CARRY METHOD BETWEEN ALU's
	USING '181 AND '182	USING 'LS181 AND '182	USING 'S181 AND 'S182	ARITHMETIC/ LOGIC UNITS	LOOK-AHEAD CARRY GENERATORS	
1 to 4	24 ns	24 ns	11 ns	1		NONE
5 to 8	36 ns	40 ns	18 ns	2		RIPPLE
9 to 16	36 ns	44 ns	19 ns	3 or 4	1	FULL LOOK-AHEAD
17 to 64	60 ns	68 ns	28 ns	5 to 16	2 to 5	FULL LOOK-AHEAD

Is snelheid niet belangrijk, dan kunnen de ripple-carry input (C_n) en ripple-carry output (C_{n+4}) worden gebruikt.

De ALU kan actief-HOGE of actief-LAGE data verwerken, waarbij de pen-functies als volgt worden geïnterpreteerd:

Deel 6: Digitale rekenkundige schakelingen

PIN NUMBER	2	1	23	22	21	20	19	18	9	10	11	13	7	16	15	17
Active-low data (Table 1)	\bar{A}_0	\bar{B}_0	\bar{A}_1	\bar{B}_1	\bar{A}_2	\bar{B}_2	\bar{A}_3	\bar{B}_3	F_0	F_1	F_2	F_3	C_n	C_{n+4}	P	G
Active-high data (Table 2)	A_0	B_0	A_1	B_1	A_2	B_2	A_3	B_3	F_0	F_1	F_2	F_3	C_n	C_{n+4}	X	Y

De ALU kan ook dienen als comparator (ALU in de subtract mode, $C_n = H$):

INPUT C_n	OUTPUT C_{n+4}	ACTIVE-LOW DATA (FIGURE 1)	ACTIVE-HIGH DATA (FIGURE 2)
H	H	$A > B$	$A < B$
H	L	$A < B$	$A > B$
L	H	$A > B$	$A < B$
L	L	$A < B$	$A > B$

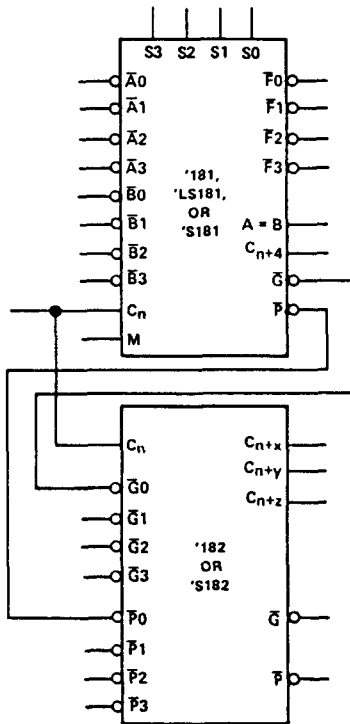


FIGURE 1
(Use with Table 1)

Toepassing 1 (aktief-LAAG)

SELECTION S3 S2 S1 S0	M = H LOGIC FUNCTIONS	ACTIVE-LOW DATA	
		M = L: ARITHMETIC OPERATIONS	
		$C_n = L$ (no carry)	$C_n = H$ (with carry)
L L L L	$F = \bar{A}$	$F = A \text{ MINUS } 1$	$F = A$
L L L H	$F = \bar{A} \oplus B$	$F = AB \text{ MINUS } 1$	$F = AB$
L L H L	$F = \bar{A} \oplus B$	$F = \bar{A} \oplus B \text{ MINUS } 1$	$F = A \oplus B$
L L H H	$F = 1$	$F = \text{MINUS } 1 \text{ (2's COMPL)}$	$F = \text{ZERO}$
L H L L	$F = \bar{A} \oplus B$	$F = A \text{ PLUS } (A \oplus B)$	$F = A \text{ PLUS } (A \oplus B) \text{ PLUS } 1$
L H L H	$F = \bar{B}$	$F = AB \text{ PLUS } (A \oplus B)$	$F = AB \text{ PLUS } (A \oplus B) \text{ PLUS } 1$
L H H L	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L H H H	$F = A \oplus B$	$F = A \oplus B$	$F = (A \oplus B) \text{ PLUS } 1$
H L L L	$F = \bar{A} \oplus B$	$F = A \text{ PLUS } (A \oplus B)$	$F = A \text{ PLUS } (A \oplus B) \text{ PLUS } 1$
H L L H	$F = A \oplus B$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H L H L	$F = B$	$F = \bar{A} \oplus B \text{ PLUS } (A \oplus B)$	$F = \bar{A} \oplus B \text{ PLUS } (A \oplus B) \text{ PLUS } 1$
H L H H	$F = A \oplus B$	$F = (A \oplus B)$	$F = (A \oplus B) \text{ PLUS } 1$
H H L L	$F = 0$	$F = A \text{ PLUS } A^*$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H H L H	$F = \bar{A} \oplus B$	$F = AB \text{ PLUS } A$	$F = AB \text{ PLUS } A \text{ PLUS } 1$
H H H L	$F = AB$	$F = \bar{A} \oplus B \text{ PLUS } A$	$F = \bar{A} \oplus B \text{ PLUS } A \text{ PLUS } 1$
H H H H	$F = A$	$F = A$	$F = A \text{ PLUS } 1$

*Each bit is shifted to the next more significant position.

tabel 1

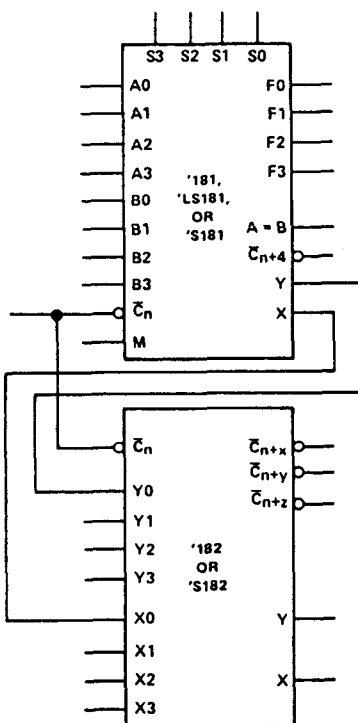


FIGURE 2
(Use with Table 2)

Toepassing 2 (aktief-HOOG)

SELECTION S3 S2 S1 S0	M = H LOGIC FUNCTIONS	ACTIVE-HIGH DATA	
		M = L: ARITHMETIC OPERATIONS	
		$C_n = H$ (no carry)	$C_n = L$ (with carry)
L L L L	$F = \bar{A}$	$F = A$	$F = A \text{ PLUS } 1$
L L L H	$F = \bar{A} \oplus B$	$F = A \oplus B$	$F = (A \oplus B) \text{ PLUS } 1$
L L H L	$F = \bar{A} \oplus B$	$F = A \oplus B$	$F = (A \oplus B) \text{ PLUS } 1$
L L H H	$F = 0$	$F = \text{MINUS } 1 \text{ (2's COMPL)}$	$F = \text{ZERO}$
L H L L	$F = \bar{A} \oplus B$	$F = A \text{ PLUS } A \oplus B$	$F = A \text{ PLUS } A \oplus B \text{ PLUS } 1$
L H L H	$F = \bar{B}$	$F = (A \oplus B) \text{ PLUS } A \oplus B$	$F = (A \oplus B) \text{ PLUS } A \oplus B \text{ PLUS } 1$
L H H L	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L H H H	$F = \bar{A} \oplus B$	$F = \bar{A} \oplus B \text{ MINUS } 1$	$F = \bar{A} \oplus B$
H L L L	$F = \bar{A} \oplus B$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB \text{ PLUS } 1$
H L L H	$F = A \oplus B$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H L H L	$F = B$	$F = (A \oplus B) \text{ PLUS } AB$	$F = (A \oplus B) \text{ PLUS } AB \text{ PLUS } 1$
H L H H	$F = AB$	$F = AB \text{ MINUS } 1$	$F = AB$
H H L L	$F = 1$	$F = A \text{ PLUS } A^*$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H H L H	$F = \bar{A} \oplus B$	$F = (A \oplus B) \text{ PLUS } A$	$F = (A \oplus B) \text{ PLUS } A \text{ PLUS } 1$
H H H L	$F = A \oplus B$	$F = (A \oplus B) \text{ PLUS } A$	$F = (A \oplus B) \text{ PLUS } A \text{ PLUS } 1$
H H H H	$F = A$	$F = A \text{ MINUS } 1$	$F = A$

tabel 2

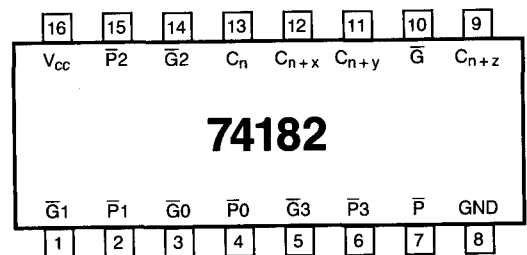
6.2 74xx-serie TTL en HC

74182

Look-ahead Carry-generator

(carry-functies compatibel met ALU)

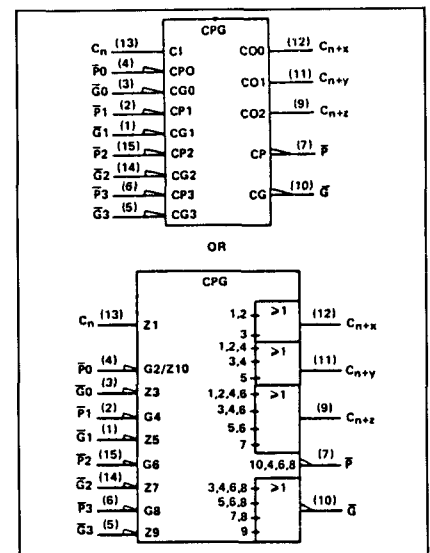
Figuur 6/6.2-182



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
$I_{CC} \begin{smallmatrix} H \\ L \end{smallmatrix}$	$\begin{smallmatrix} 27 \\ 45 \end{smallmatrix}$		$\begin{smallmatrix} 18.4 \\ 23.5 \end{smallmatrix}$	$\begin{smallmatrix} 35 \\ 69 \end{smallmatrix}$		$\begin{smallmatrix} 17 \\ 23 \end{smallmatrix}$			80^5	mA
I_{OS}	$\begin{smallmatrix} -40 \\ -100 \end{smallmatrix}$		$\begin{smallmatrix} -60 \\ -150 \end{smallmatrix}$	$\begin{smallmatrix} -40 \\ -100 \end{smallmatrix}$		$\begin{smallmatrix} -30 \\ -112 \end{smallmatrix}$			4	mA
T_{plh}/T_{phl}^1	$\begin{smallmatrix} 11/ \\ 15 \end{smallmatrix}$		$\begin{smallmatrix} 6.2/ \\ 3.7 \end{smallmatrix}$	$\begin{smallmatrix} 4.5/ \\ 4.5 \end{smallmatrix}$		$\begin{smallmatrix} 5/ \\ 5 \end{smallmatrix}$			25	ns
T_{plh}/T_{phl}^2	$\begin{smallmatrix} 11/ \\ 15 \end{smallmatrix}$		$\begin{smallmatrix} 7.9/ \\ 6 \end{smallmatrix}$	$\begin{smallmatrix} 5/ \\ 7 \end{smallmatrix}$		$\begin{smallmatrix} 6/ \\ 5 \end{smallmatrix}$			25	ns
T_{plh}/T_{phl}^3	$\begin{smallmatrix} 11/ \\ 15 \end{smallmatrix}$		$\begin{smallmatrix} 5.7/ \\ 4.1 \end{smallmatrix}$	$\begin{smallmatrix} 4.5/ \\ 6.5 \end{smallmatrix}$		$\begin{smallmatrix} 5/ \\ 5 \end{smallmatrix}$			18	ns
T_{plh}/T_{phl}^4	$\begin{smallmatrix} 11/ \\ 15 \end{smallmatrix}$		$\begin{smallmatrix} 6.6/ \\ 6.8 \end{smallmatrix}$	$\begin{smallmatrix} 6.5/ \\ 7 \end{smallmatrix}$		$\begin{smallmatrix} 5/ \\ 5 \end{smallmatrix}$			20	ns

1) $\bar{G}0, \bar{G}1, \bar{G}2, \bar{G}3, P0, P1, P2$ of $P3 \rightarrow C_{n+x}, C_{n+y}$ of C_{n+z}
 2) idem (niet $P0$) $\rightarrow \bar{G}$

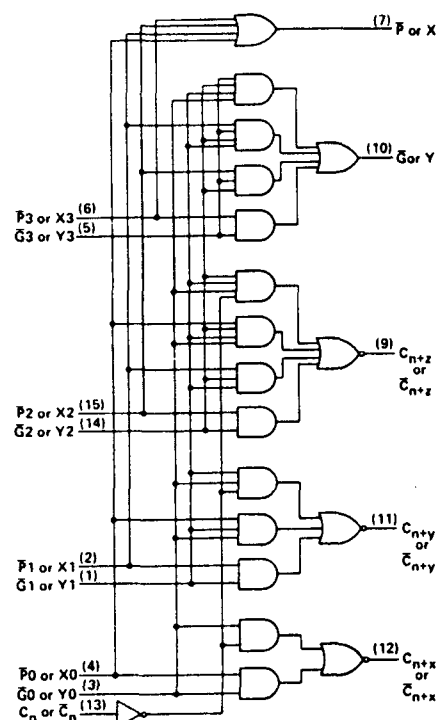
3) $\bar{P}0, \bar{P}1, \bar{P}2$ of $\bar{P}3 \rightarrow \bar{P}$
 4) $C_n \rightarrow C_{n+x}, C_{n+y}$ of C_{n+z}
 5) μA



logische symbolen

PIN DESIGNATIONS		
ALTERNATIVE	DESIGNATIONS ¹	FUNCTION
$\bar{G}0, \bar{G}1, \bar{G}2, \bar{G}3$	$G0, G1, G2, G3$	Carry Generate Inputs
$\bar{P}0, \bar{P}1, \bar{P}2, \bar{P}3$	$P0, P1, P2, P3$	Carry Propagate Inputs
C_n	C_n	Carry Input
$C_{n+x}, C_{n+y}, C_{n+z}$	$C_{n+x}, C_{n+y}, C_{n+z}$	Carry Outputs
\bar{G}	Y	Carry Generate Output
\bar{P}	X	Carry Propagate Output
V_{CC}		Supply Voltage
GND		Ground

¹ Interpretations are illustrated in connection with the Function Tables for the 'HC181 and 'HC881.



functioneel blokschema (pos. logica)

Deel 6: Digitale rekenkundige schakelingen

$$\begin{aligned}
 C_{n+x} &= G_0 + P_0 C_n \\
 C_{n+y} &= G_1 + P_1 G_0 + P_1 P_0 C_n \\
 C_{n+z} &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_n \\
 \bar{G} &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 \\
 \bar{P} &= P_3 P_2 P_1 P_0
 \end{aligned}$$

$$\begin{aligned}
 \bar{C}_{n+x} &= \overline{Y_0 (X_0 + C_n)} \\
 \bar{C}_{n+y} &= \overline{Y_1 (X_1 + Y_0 (X_0 + C_n))} \\
 \bar{C}_{n+z} &= \overline{Y_2 (X_2 + Y_1 (X_1 + Y_0 (X_0 + C_n)))} \\
 Y &= Y_3 (X_3 + Y_2) (X_3 + X_2 + Y_1) (X_3 + X_2 + X_1 + Y_0) \\
 X &= X_3 + X_2 + X_1 + X_0
 \end{aligned}$$

logische vergelijkingen voor de 74182

FUNCTION TABLE FOR \bar{G} OUTPUT							OUTPUT \bar{G}
\bar{G}_3	\bar{G}_2	\bar{G}_1	\bar{G}_0	\bar{P}_3	\bar{P}_2	\bar{P}_1	
L	X	X	X	X	X	X	L
X	L	X	X	L	X	X	L
X	X	L	X	L	L	X	L
X	X	X	L	L	L	L	L
All other combinations							H

INPUTS				OUTPUT
\bar{P}_3	\bar{P}_2	\bar{P}_1	\bar{P}_0	\bar{P}
L	L	L	L	L
All other combinations				H

FUNCTION TABLE FOR C_{n+x} OUTPUT			
\bar{G}_0	\bar{P}_0	C_n	C_{n+x}
L	X	X	H
X	L	H	H
All other combinations			L

FUNCTION TABLE C_{n+y} OUTPUT					
\bar{G}_1	\bar{G}_0	\bar{P}_1	\bar{P}_0	C_n	C_{n+y}
L	X	X	X	X	H
X	L	L	X	X	H
X	X	L	L	H	H
All other combinations					L

FUNCTION TABLE FOR C_{n+z} OUTPUT							
\bar{G}_2	\bar{G}_1	\bar{G}_0	\bar{P}_2	\bar{P}_1	\bar{P}_0	C_n	C_{n+z}
L	X	X	X	X	X	X	H
X	L	X	L	X	X	X	H
X	X	L	L	L	X	X	H
X	X	X	L	L	L	H	H
All other combinations							L

H = High-level, L = Low-level, X = Irrelevant
 Any inputs not shown in a given table are irrelevant with respect to that output.

waarheidstabellen

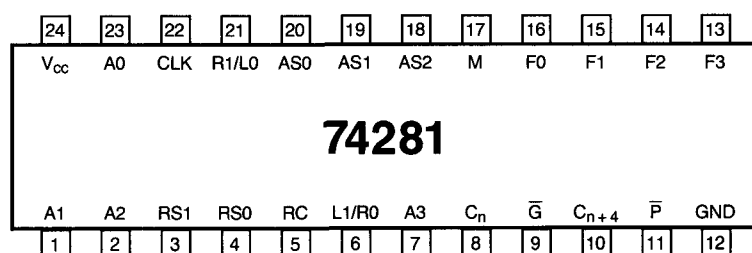
6.2 74xx-serie TTL en HC

74281

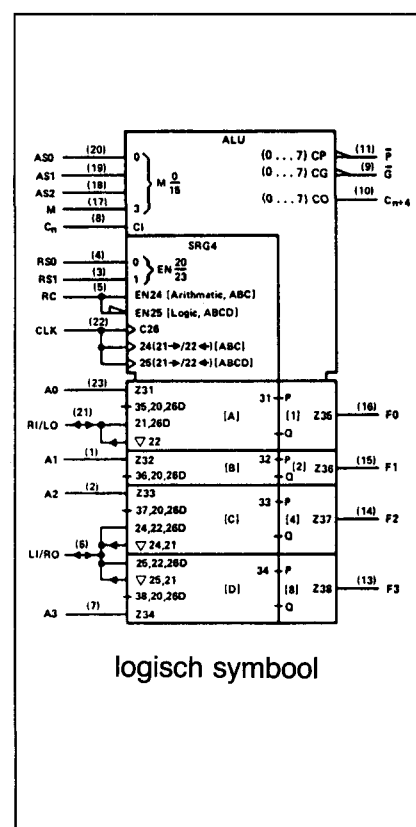
4-bit parallelle binaire accumulator

Figuur 6/6.2-281

(ALU + schuif-/opslag matrix)



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
Icc				144						mA
Ios				-40 -110						mA
Tplh/ Tphl ¹⁾				10/ 10						ns
Tplh/ Tphl ²⁾				18/ 18						ns
Tplh/ Tphl ³⁾				10/ 10						ns
Tplh/ Tphl ⁴⁾				14/ 14						ns
Tplh/ Tphl ⁵⁾				30/ 30						ns
Tplh/ Tphl ⁶⁾				7/ 7						ns
Tplh/ Tphl ⁷⁾				28/ 28						ns
Tplh/ Tphl ⁸⁾				30/ 30						ns

1) $C_n \rightarrow C_{n+4}$ 2) $A \rightarrow C_{n+4}$ 3) $C_n \rightarrow F$ 4) $A \rightarrow \bar{G}$ 5) $A0, A3 \rightarrow R1/L0, \text{ resp. } L1/R0$ 6) $F0, F3 \rightarrow R1/L0, \text{ resp. } L1/R0$ 7) $AS \rightarrow F \text{ of } C_{n+4}$ 8) clock $\rightarrow F$ 

Deel 6: Digitale rekenkundige schakelingen

waarheidstabellen

TABLE 1—ARITHMETIC FUNCTIONS
Mode Control (M) = Low

ALU SELECTION			ACTIVE-HIGH DATA	
AS2	AS1	AS0	$C_n = H$ (with carry)	$C_n = L$ (no carry)
L	L	L	$F_0 = L, F_1 = F_2 = F_3 = H$	$F_n = H$
L	L	H	$F = B \text{ MINUS } A$	$F = B \text{ MINUS } A \text{ MINUS } 1$
L	H	L	$F = A \text{ MINUS } B$	$F = A \text{ MINUS } B \text{ MINUS } 1$
L	H	H	$F = A \text{ PLUS } B \text{ PLUS } 1$	$F = A \text{ PLUS } B$
H	L	L	$F = B \text{ PLUS } 1$	$F_n = B_n$
H	L	H	$F = \bar{B} \text{ PLUS } 1$	$F_n = \bar{B}_n$
H	H	L	$F = A \text{ PLUS } 1$	$F_n = A_n$
H	H	H	$F = \bar{A} \text{ PLUS } 1$	$F_n = \bar{A}_n$

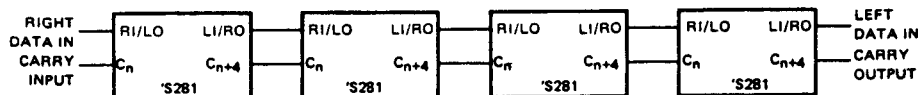
TABLE 2—LOGIC FUNCTIONS
Mode Control (M) = High
Carry Input (C_n) = X (Irrelevant)

ALU SELECTION			ACTIVE-HIGH DATA FUNCTION
AS2	AS1	AS0	
L	L	L	$F_n = L$
L	X	H	$F_n = A_n \oplus B_n$
L	H	L	$F_n = A_n \odot B_n$
H	L	L	$F_n = \frac{A_n B_n}{A_n + B_n}$
H	L	H	$F_n = A_n + B_n$
H	H	L	$F_n = A_n B_n$
H	H	H	$F_n = A_n + B_n$

TABLE 3—SHIFT-MODE FUNCTIONS
 $C_n = M = AS0 = AS1 = L$, and $AS2 = H$ ($F_n = B_n$)

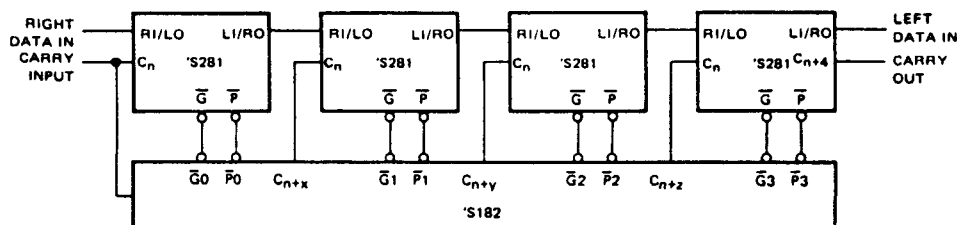
FUNCTION	INPUTS BEFORE ↑									CLOCK INPUT	OUTPUTS AFTER ↑								
	REGISTER SELECTION RS0 RS1		REGISTER CONTROL INPUT	INPUT/ OUTPUT RI/LO	SHIFT-MATRIX INPUTS				INPUT/ OUTPUT LI/RO		INPUT/ OUTPUT RI/LO	SHIFT-MATRIX OUTPUTS (ALU 8 INPUTS)				INPUT/ OUTPUT LI/RO			
					F0	F1	F2	F3				Q _A	Q _B	Q _C	Q _D				
LOAD	L	L	X	Z	f0	f1	f2	f3	Z	↑	Z	f0	f1	f2	f3	Z			
LSL	L	H	L	Q _A	Q _A	Q _B	Q _C	Q _D	li	↑	Q _{Bn}	Q _{Bn}	Q _{Cn}	Q _{Dn}	li	li			
LSA	L	H	H	Q _A	Q _A	Q _B	Q _C	Q _D	li	↑	Q _{Bn}	Q _{Bn}	Q _{Cn}	Q _{Dn}	li	li			
RSL	H	L	L	ri	Q _A	Q _B	Q _C	Q _D	Q _D	↑	ri	ri	Q _{An}	Q _{Bn}	Q _{Cn}	Q _{Cn}			
RSA	H	L	H	ri	Q _A	Q _B	Q _C	Q _D	Q _C	↑	ri	ri	Q _{An}	Q _{Bn}	Q _{D0}	Q _{Bn}			
HOLD	H	H	X	X	Q _A	Q _B	Q _C	Q _D	X	↑	Z	Q _{A0}	Q _{B0}	Q _{C0}	Q _{D0}	Z			
	X	X	X	X	Q _A	Q _B	Q _C	Q _D	X	L	RI/LO	Q _{A0}	Q _{B0}	Q _{C0}	Q _{D0}	LI/RO			

H = high level (steady state)
 L = low level (steady state)
 X = irrelevant (any input, including transitions)
 Z = high impedance (output off)
 \uparrow = transition from low to high level
 f0, f1, f2, f3, ri, li = the level of steady-state conditions at F0, F1, F2, F3, RI/LO, or LI/RO respectively
 QA0, QB0, QC0, QD0 = the level of QA, QB, QC, or QD, respectively, before the indicated steady-state input conditions were established
 QA_n, QB_n, QC_n, QD_n = the level of QA, QB, QC, or QD, respectively, before the most recent transition of the clock



ENTER AND STORE TIME: 38 ns typical
 EACH SUCCESSIVE ADDITION TO STORED DATA: 44 ns typical

16-BIT BINARY ACCUMULATOR USING FOUR SN54S281/SN74S281 CIRCUITS
 IN RIPPLE-CARRY MODE



ENTER AND STORE TIME: 37 ns typical
 EACH SUCCESSIVE ADDITION TO STORED DATA: 29 ns typical

16-BIT BINARY ACCUMULATOR USING FOUR SN54S281/SN74S281 CIRCUITS
 AND ONE SN54S182/SN74S182 IN FULL LOOK-AHEAD CARRY MODE

toepassingen

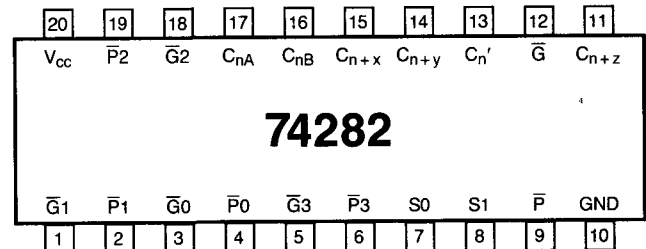
6.2 74xx-serie TTL en HC

74282

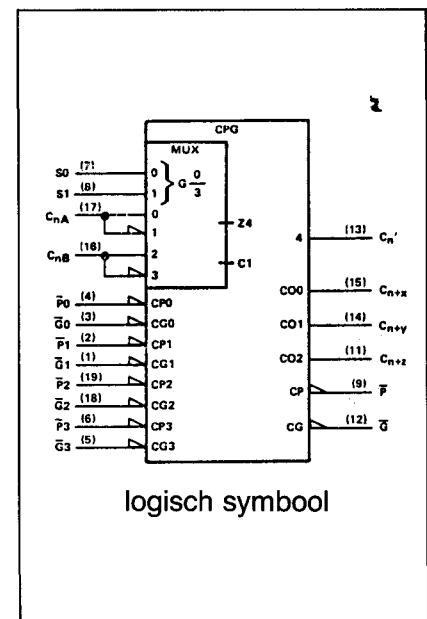
Look-ahead carry-generator met selecteerbare carry-ingangen

(carry-functies compatibel met ALU)

Figuur 6/6.2-282



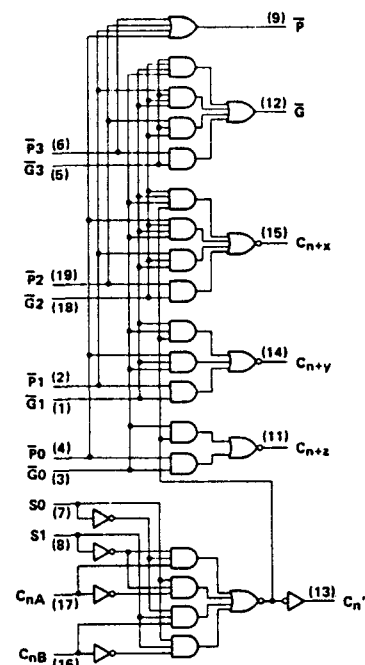
LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABLE PARAMETERS										Eenheid
I_{cc} $\begin{smallmatrix} H \\ L \end{smallmatrix}$						22 26				mA
I_{os}						-30 -112				mA
T_{plh}/T_{phl} ¹⁾						6/ 6				ns
T_{plh}/T_{phl} ²⁾						6/ 6				ns
T_{plh}/T_{phl} ³⁾						5/ 5				ns
T_{plh}/T_{phl} ⁴⁾						6/ 5				ns
T_{plh}/T_{phl} ⁵⁾						5/ 5				ns



- 1) S0, S1, CnA of CnB \rightarrow Cn'
 2) S0, S1, CnA of CnB \rightarrow Cn+x, Cn+y, Cn+z
 3) P-bar of G \rightarrow Cn+x, Cn+y, Cn+z
 4) P-bar of G \rightarrow G-bar
 5) P-bar \rightarrow P-bar

PIN DESIGNATIONS		
ALTERNATIVE DESIGNATIONS†		FUNCTION
$\bar{G}_0, \bar{G}_1, \bar{G}_2, \bar{G}_3$	G0, G1, G2, G3	Carry Generate Inputs
$\bar{P}_0, \bar{P}_1, \bar{P}_2, \bar{P}_3$	P0, P1, P2, P3	Carry Propagate Inputs
C_{nA}, C_{nB}	$\bar{C}_{nA}, \bar{C}_{nB}$	Carry Inputs
C_n	\bar{C}_n	Selected Carry
$C_{n+x}, C_{n+y}, C_{n+z}$	$\bar{C}_{n+x}, \bar{C}_{n+y}, \bar{C}_{n+z}$	Carry Outputs
\bar{G}	Y	Carry Generate Outputs
\bar{P}	X	Carry Propagate Outputs
S0, S1		Carry Select Inputs
VCC		Supply Voltage
GND		Ground

† Interpretations are illustrated in connection with the Function Tables for the 'AS181A and 'AS881A.



Deel 6: Digitale rekenkundige schakelingen

$$\begin{aligned}
 C_{n+x} &= G_0 + P_0 C_n \\
 C_{n+y} &= G_1 + P_1 G_0 + P_1 P_0 C_n \\
 C_{n+z} &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_n \quad \text{or} \quad \bar{C}_{n+x} = \bar{Y}_0 (X_0 + \bar{C}_n) \\
 &\quad \bar{C}_{n+y} = \bar{Y}_1 (X_1 + Y_0 (X_0 + \bar{C}_n)) \\
 &\quad \bar{C}_{n+z} = \bar{Y}_2 (X_2 + Y_1 (X_1 + Y_0 (X_0 + \bar{C}_n))) \\
 &\quad \bar{G} = \bar{G}_3 + \bar{P}_3 \bar{G}_2 + \bar{P}_3 \bar{P}_2 \bar{G}_1 + \bar{P}_3 \bar{P}_2 \bar{P}_1 \bar{G}_0 \quad Y = Y_3 (X_3 + Y_2 (X_3 + X_2 + Y_1 (X_3 + X_2 + X_1 + Y_0))) \\
 &\quad \bar{P} = \bar{P}_3 \bar{P}_2 \bar{P}_1 \bar{P}_0 \quad X = X_3 + X_2 + X_1 + X_0
 \end{aligned}$$

logische vergelijkingen voor de 74282

FUNCTION TABLE FOR \bar{G} OUTPUT

INPUTS							OUTPUT
\bar{G}_3	\bar{G}_2	\bar{G}_1	\bar{G}_0	\bar{P}_3	\bar{P}_2	\bar{P}_1	
L	X	X	X	X	X	X	L
X	L	X	X	L	X	X	L
X	X	L	X	L	L	X	L
X	X	X	L	L	L	L	L
All other combinations							H

FUNCTION TABLE FOR \bar{P} OUTPUT

INPUTS				OUTPUT
\bar{P}_3	\bar{P}_2	\bar{P}_1	\bar{P}_0	
L	L	L	L	L
All other combinations				H

FUNCTION TABLE

FOR C_n OUTPUT

INPUTS		OUTPUT
S1	S0	
L	L	C_{nA}
L	H	C_{nA}
H	L	C_{nB}
H	H	C_{nB}

FUNCTION TABLE
FOR C_{n+x} OUTPUT

INPUTS			OUTPUT
\bar{G}_0	\bar{P}_0	C_n	
L	X	X	H
X	L	H	H
All other combinations			L

FUNCTION TABLE C_{n+y} OUTPUT

INPUTS					OUTPUT
\bar{G}_1	\bar{G}_0	\bar{P}_1	\bar{P}_0	C_n	
L	X	X	X	X	H
X	L	L	X	X	H
X	X	L	L	H	H
All other combinations					L

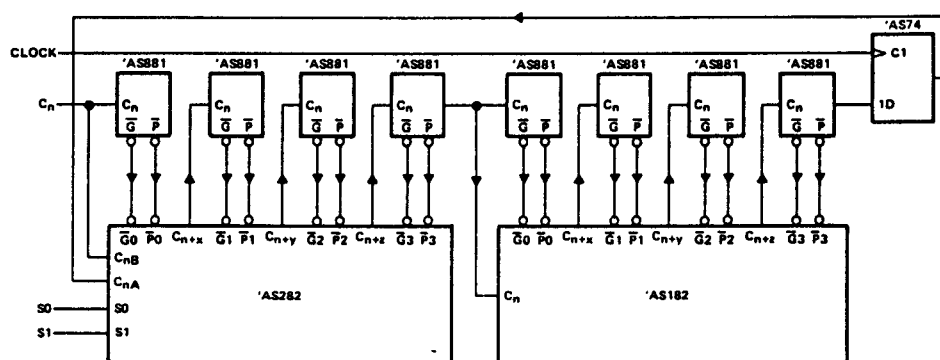
FUNCTION TABLE FOR C_{n+z} OUTPUT

INPUTS							OUTPUT
\bar{G}_2	\bar{G}_1	\bar{G}_0	\bar{P}_2	\bar{P}_1	\bar{P}_0	C_n	
L	X	X	X	X	X	X	H
X	L	X	L	X	X	X	H
X	X	L	L	L	X	X	H
X	X	X	L	L	L	H	H
All other combinations							L

H = high-level, L = low level, X = irrelevant.

Any inputs not shown in a given table are irrelevant with respect to that output.

waarheidstabellen



32-BIT LOOK-AHEAD CARRY WITH DOUBLE-PRECISION CARRY IN 'AS282 AND 'AS182

toepassingsvoorbeeld

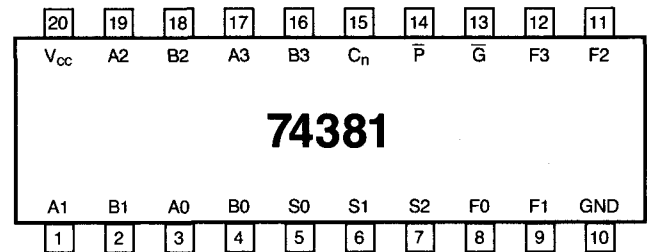
6.2 74xx-serie TTL en HC

74381

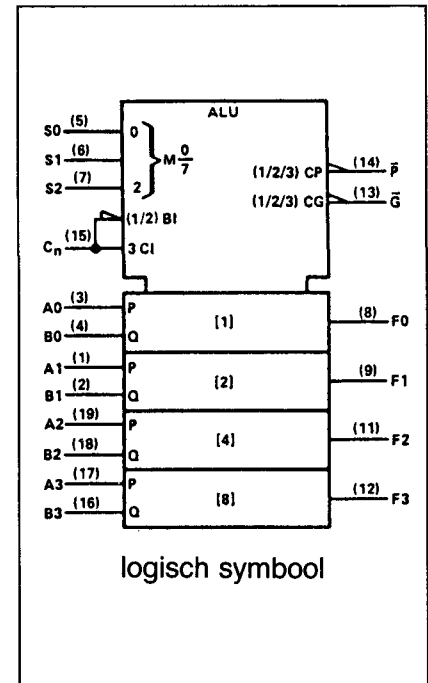
Arithmetic Logic Unit

ALU/functiegenerator met \bar{G} en \bar{P} uitgangen
voor look-ahead carry cascade

Figuur 6/6.2-381



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
Icc			59	105	35				80 ⁸⁾	mA
Ios			-60 -150	-40 -100	-20 -100				4	mA
Tplh/ Tphl ²⁾			8.1/ 5.7	10/ 10	18/ 14				12	ns
Tplh/ Tphl ³⁾			6.4/ 6.8	12/ 12	20/ 21				12	ns
Tplh/ Tphl ⁴⁾			7.2/ 6.5	11/ 11	21/ 23				12	ns
Tplh/ Tphl ⁵⁾			10.4/ 8.2	18/ 16	20/ 15				12	ns
Tplh/ Tphl ⁶⁾			8.3/ 8.2	18/ 18	35/ 34				17	ns
Tplh/ Tphl ⁷⁾			7.8/ 10.2	18/ 18	31/ 32				17	ns



1) 74LS381A

2) $C_n \rightarrow F$ 3) $A \text{ of } B \rightarrow \bar{G}$ 4) $A \text{ of } B \rightarrow \bar{P}$ 5) $A_i \text{ of } B_i \rightarrow F_i$ 6) $S_i \rightarrow F_i$ 7) $S_i \rightarrow \bar{G} \text{ of } \bar{P}$ 8) μA

FUNCTION TABLE

SELECTION	ARITHMETIC/LOGIC		
S2 S1 S0	OPERATION		
L L L	CLEAR		
L L H	B MINUS A		
L H L	A MINUS B		
L H H	A PLUS B		
H L L	$A \oplus B$		
H L H	$A + B$		
H H L	AB		
H H H	PRESET		

H = high level, L = low level

waarheidstabel 1

FUNCTION TABLE

ARITHMETIC/LOGIC OPERATION	INPUTS						OUTPUTS				('HC381)	
	S2	S1	S0	C _n	A _n	B _n	F3	F2	F1	F0	\bar{G}	\bar{P}
Clear	L	L	L	X	X	X	L	L	L	L	H	H
B MINUS A	L	L	H	L	L	L	H	H	H	H	H	L
				L	L	H	H	H	H	L	L	H
				L	H	L	L	L	L	L	H	H
				L	H	H	H	H	H	H	H	L
				H	L	L	L	L	L	L	H	L
				H	L	H	H	H	H	H	L	H
				H	H	L	L	L	L	H	H	H
				H	H	H	L	L	L	L	H	L
A MINUS B	L	H	L	L	L	L	H	H	H	H	H	L
				L	L	H	L	L	L	L	H	H
				L	H	L	H	H	H	L	L	H
				L	H	H	H	H	H	H	H	L
				H	L	L	L	L	L	L	H	L
				H	L	H	L	L	L	H	H	H
				H	H	L	H	H	H	H	L	H
				H	H	H	L	L	L	L	H	L
A PLUS B	L	H	H	L	L	L	L	L	L	L	H	H
				L	L	H	H	H	H	H	H	L
				L	H	L	H	H	H	H	H	L
				L	H	H	H	H	H	L	L	H
				H	L	L	L	L	L	H	H	H
				H	L	H	L	L	L	L	H	L
				H	H	L	L	L	L	L	H	L
				H	H	H	H	H	H	H	L	H
$A \oplus B$	H	L	L	X	L	L	L	L	L	L	H	H
				L	L	H	H	H	H	H	H	L
				H	L	H	H	H	H	H	H	L
				L	H	L	H	H	H	H	H	L
				H	H	L	H	H	H	H	H	L
				X	H	H	L	L	L	L	H	H
				L	L	L	L	L	L	L	H	L
				L	L	H	H	H	H	H	H	L
A + B	H	L	H	L	L	L	L	L	L	L	H	H
				L	L	H	H	H	H	H	H	L
				L	H	L	H	H	H	H	H	L
				L	H	H	H	H	H	H	H	L
				H	L	L	L	L	L	L	H	H
				H	L	H	H	H	H	H	H	L
				H	H	L	H	H	H	H	H	L
				H	H	H	H	H	H	H	H	L
AB	H	H	L	X	L	L	L	L	L	L	H	H
				X	L	H	L	L	L	L	H	H
				X	H	L	L	L	L	L	H	H
				L	H	H	H	H	H	H	H	L
				H	H	H	H	H	H	H	H	L
				L	L	L	L	L	L	L	H	H
				L	L	H	H	H	H	H	H	L
				L	L	H	H	H	H	H	H	L
PRESET	H	H	H	L	X	X	H	H	H	H	H	L
				H	X	X	H	H	H	H	H	L

waarheidstabel 2

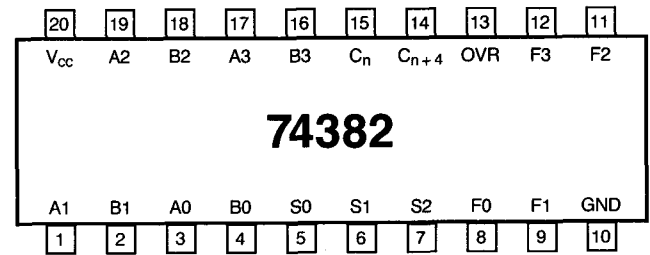
6.2 74xx-serie TTL en HC

74382

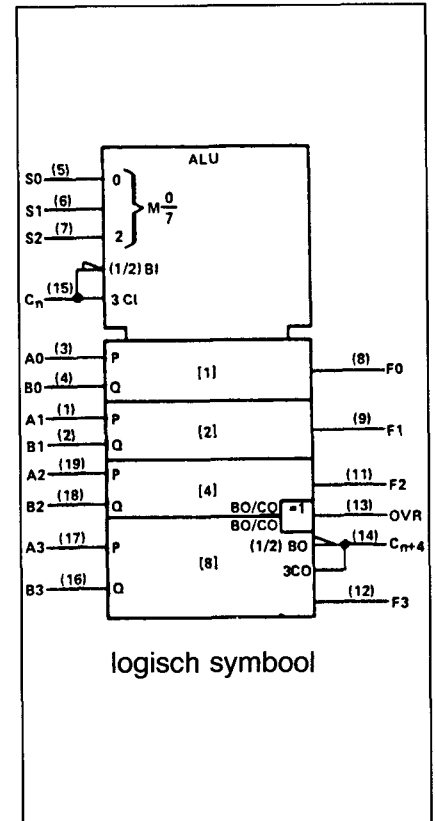
Arithmetic Logic Unit

ALU/functiegenerator met ripple-carry en overflow uitgangen

Figuur 6/6.2-382



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
I _{cc}			54		35				80 ⁹⁾	mA
I _{os}			-60 -150		-20 -100				4	mA
T _{plh} / T _{phl} ¹⁾			8.1/ 5.7		18/ 14				12	ns
T _{plh} / T _{phl} ²⁾			10.4/ 8.2		20/ 15				12	ns
T _{plh} / T _{phl} ³⁾			11/ 8.2		35/ 34				17	ns
T _{plh} / T _{phl} ⁴⁾			6/ 6.5		28/ 26				13	ns
T _{plh} / T _{phl} ⁵⁾			12.5/ 9		23/ 27				14	ns
T _{plh} / T _{phl} ⁶⁾			11.5/ 8		38/ 36				20	ns
T _{plh} / T _{phl} ⁷⁾			8/ 7.1		10/ 13				12	ns
T _{plh} / T _{phl} ⁸⁾			5.6/ 6.3		13/ 11				11	ns



- 1) C_n → F
 2) A_i of B_i → F_i
 3) S_i → F_i
 4) A of B → C_{n+4}
 5) A of B → OVR
 6) S_i → C_{n+4} of OVR

- 7) C_n → OVR
 8) C_n → C_{n+4}
 9) μA

FUNCTION TABLE

SELECTION	ARITHMETIC/LOGIC
S2 S1 S0	OPERATION
L L L	CLEAR
L L H	B MINUS A
L H L	A MINUS B
L H H	A PLUS B
H L L	A ⊕ B
H L H	A + B
H H L	AB
H H H	PRESET

H = high level, L = low level

waarheidstabel 1

FUNCTION TABLE

ARITHMETIC/LOGIC OPERATION	INPUTS						OUTPUTS				('HC382)	
	S2	S1	S0	C _n	A _n	B _n	F3	F2	F1	F0	OVR	C _{n+4}
Clear	L	L	L	X	X	X	L	L	L	L	L	L
B MINUS A	L	L	H	L	L	L	H	H	H	H	L	L
				L	L	H	H	H	H	L	L	H
				L	H	L	L	L	L	L	L	L
				L	H	H	H	H	H	H	L	L
				H	L	L	L	L	L	L	L	H
				H	L	H	H	H	H	H	L	H
				H	H	L	L	L	L	H	L	L
				H	H	H	L	L	L	L	L	H
A MINUS B	L	H	L	L	L	L	H	H	H	H	L	L
				L	L	H	L	L	L	L	L	H
				L	H	L	H	H	H	L	L	L
				L	H	H	H	H	H	H	L	L
				H	L	L	L	L	L	L	L	H
				H	L	H	L	L	L	H	L	L
				H	H	L	H	H	H	H	L	H
				H	H	H	L	L	L	L	L	H
A PLUS B	L	H	H	L	L	L	H	L	L	L	L	L
				L	L	H	H	H	H	H	L	L
				L	H	L	H	H	H	H	L	L
				L	H	H	H	H	H	L	L	H
				H	L	L	L	L	L	H	L	L
				H	L	H	L	L	L	L	L	H
				H	H	L	L	L	L	L	L	H
				H	H	H	H	H	H	H	L	H
A ⊕ B	H	L	L	X	L	L	L	L	L	L	L	L
				L	L	H	H	H	H	H	L	L
				H	L	H	H	H	H	H	H	L
				L	H	L	H	H	H	H	L	L
				H	H	L	H	H	H	H	H	L
A + B	H	L	H	X	L	L	L	L	L	L	L	L
				L	L	H	H	H	H	H	L	L
				L	H	L	H	H	H	H	H	H
				H	L	L	H	H	H	H	L	L
				H	H	L	H	H	H	H	H	L
AB	H	H	L	X	L	L	L	L	L	L	L	L
				X	H	L	L	L	L	L	L	L
				L	H	H	H	H	H	H	L	L
				H	H	H	H	H	H	H	H	H
PRESET	H	H	H	L	X	X	H	H	H	H	L	L
				H	X	X	H	H	H	H	H	H

waarheidstabel 2

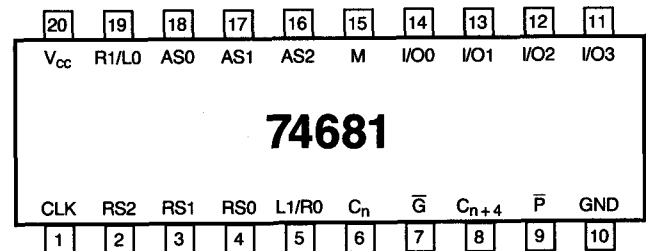
6.2 74xx-serie TTL en HC

74681

4-bit parallelle binaire accumulator

(ALU + 2 synchrone registers, I/O: bus-compatibel)

Figuur 6/6.2-681

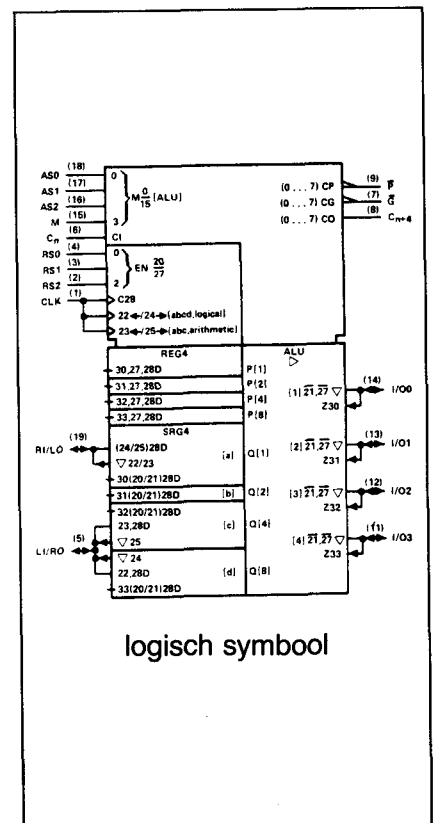


LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
Icc					100					mA
Ios					-20 -100					mA
Tplh/ Tphl ¹⁾					25/30 26/27					ns
Tplh/ Tphl ²⁾					27/29 36/34					ns
Tplh/ Tphl ³⁾					25/23 19/17					ns
Tplh/ Tphl ⁴⁾					17/ 13					ns
Tpzh/ Tpzl ⁵⁾					28/ 28					ns
Tphz/ Tplz ⁵⁾					35/ 39					ns
Tpzh/ Tpzl ⁶⁾					25/ 22					ns
Tphz/ Tplz ⁶⁾					21/ 34					ns

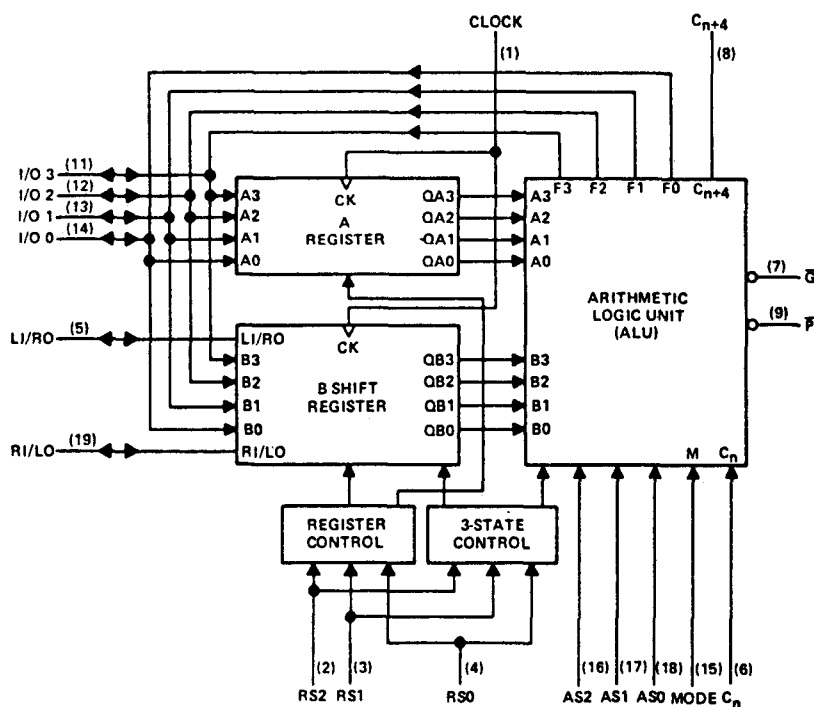
1) CLK → \bar{P}
G2) CLK → I/O
C_n+43) CLK → L1/R0
R1/L04) C_n → I/O

5) RS0 - RS2 → I/O

6) RS0 - RS2 → L1/R0



Deel 6: Digitale rekenkundige schakelingen



functioneel blokschema

TABLE 1 - ARITHMETIC FUNCTIONS

Mode Control (M) = Low

ALU SELECTION	ACTIVE-HIGH DATA	
	C _n = H (with carry)	C _n = L (no carry)
AS2 AS1 AS0		
L L L	F _j = L	F _j = H
L L H	F = B MINUS A	F = B MINUS A MINUS 1
L H L	F = A MINUS B	F = A MINUS B MINUS 1
L H H	F = A PLUS B PLUS 1	F = A PLUS B
H L L	F = B PLUS 1	F _j = B _j
H L H	F = B PLUS 1	F _j = B _j
H H L	F = A PLUS 1	F _j = A _j
H H H	F = A PLUS 1	F _j = A _j

TABLE 2 - LOGIC FUNCTIONS

Mode Control (M) = High

ALU SELECTION	ACTIVE-HIGH DATA	
	C _n = H (with carry)	C _n = L (no carry)
AS2 AS1 AS0		
L L L	F ₀ = H, F ₁ = F ₂ = F ₃ = L	F _j = L
L L H	F _j = A _j ⊕ B _j PLUS 1	F _j = A _j ⊕ B _j
L H L	F _j = A _j ⊕ B _j PLUS 1	F _j = A _j ⊕ B _j
L H H	F _j = L	F _j = H
H L L	F _j = A _j B _j PLUS 1	F _j = A _j B _j
H L H	F _j = A _j + B _j PLUS 1	F _j = A _j + B _j
H H L	F _j = A _j B _j PLUS 1	F _j = A _j B _j
H H H	F _j = A _j + B _j PLUS 1	F _j = A _j + B _j

TABLE 3 - REGISTER FUNCTIONS

FUNCTION	INPUTS BEFORE L TO H CLOCK TRANSITION										INTERNAL OUTPUTS AFTER L TO H CLOCK TRANSITION													
	REGISTER SELECTION			DATA INPUTS							A REGISTER				B SHIFT REGISTER						ALU			
	RS2	RS1	RS0	LI/RO	I/O 3	I/O 2	I/O 1	I/O 0	RI/LO	QA3	QA2	QA1	QA0	LI/RO	QB3	QB2	QB1	QB0	RI/LO	F3	F2	F1	F0	
ACCUM	L	L	L	Z	F3	F2	F1	F0	Z	QA3 ₀	QA2 ₀	QA1 ₀	QA0 ₀	Z	F3 _n	F2 _n	F1 _n	F0 _n	Z	F3	F2	F1	F0	
LOAD B	L	L	H	Z	b3	b2	b1	b0	Z	QA3 ₀	QA2 ₀	QA1 ₀	QA0 ₀	Z	b3	b2	b1	b0	Z	Z	Z	Z	Z	
LEFT SHIFT LOGICAL	L	H	L	ii	F3	F2	F1	F0	QB0	QA3 ₀	QA2 ₀	QA1 ₀	QA0 ₀	ii	ii	QB3 _n	QB2 _n	QB1 _n	QB1 _n	F3	F2	F1	F0	
LEFT SHIFT ARITH	L	H	H	ii	F3	F2	F1	F0	QB0	QA3 ₀	QA2 ₀	QA1 ₀	QA0 ₀	ii	QB3 _n	ii	QB2 _n	QB1 _n	QB1 _n	F3	F2	F1	F0	
RIGHT SHIFT LOGICAL	H	L	L	QB3	F3	F2	F1	F0	ii	QA3 ₀	QA2 ₀	QA1 ₀	QA0 ₀	QB2 _n	QB2 _n	QB1 _n	QB0 _n	ii	ii	F3	F2	F1	F0	
RIGHT SHIFT ARITH	H	L	H	QB2	F3	F2	F1	F0	ii	QA3 ₀	QA2 ₀	QA1 ₀	QA0 ₀	QB1 _n	QB3 _n	QB1 _n	QB0 _n	ii	ii	F3	F2	F1	F0	
HOLD	H	H	L	Z	F3	F2	F1	F0	Z	QA3 ₀	QA2 ₀	QA1 ₀	QA0 ₀	Z	QB3 ₀	QB2 ₀	QB1 ₀	QB0 ₀	Z	F3 ₀	F2 ₀	F1 ₀	F0 ₀	
LOAD A	H	H	H	Z	a3	a2	a1	a0	Z	a3	a2	a1	a0	Z	QB3 ₀	QB2 ₀	QB1 ₀	QB0 ₀	Z	Z	Z	Z	Z	

H = high level (steady state)

L = low level (steady state)

Z = high impedance (output off)

a0 ... a3, b0 ... b3 = the level of steady-state condition at I/O 0 thru I/O 3, respectively and intended as A or B input data

F0 ... F3 = internal ALU results

QA0₀ ... QB0₀, F0₀ ... F3₀ = the level of QA0 thru QB3 and F0 thru F3, respectively, before the indicated steady-state input conditions were establishedQA0_n ... QB3_n = the level of QA0 thru QB3 before the most recent t transition of the clock

ri, ii = the level of steady-state conditions at RI/LO or LI/RO, respectively

waarheidstabellen

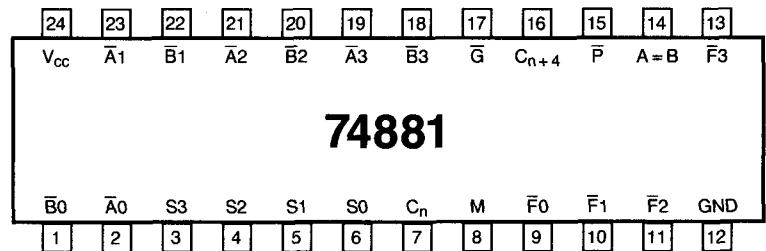
6.2 74xx-serie TTL en HC

74881

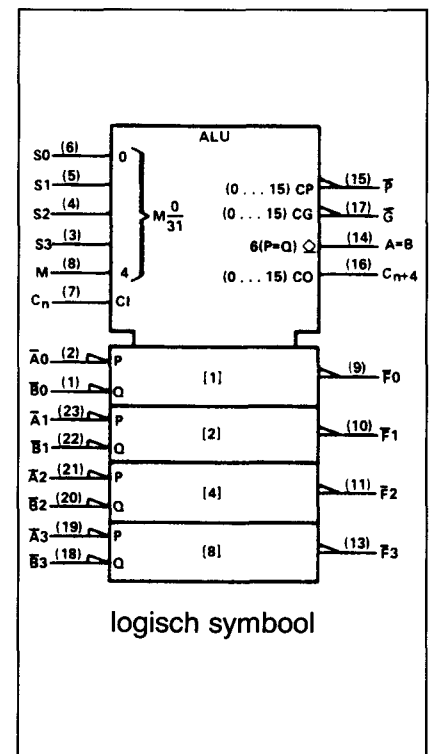
Arithmetic Logic Unit

ALU/functiegenerator
(met status register checks)

Figuur 6/6.2-881



LOGICA	TTL	L	F	S	LS	AS ¹⁾	ALS	C	HC	
VARIABELE PARAMETERS										Eenheid
Icc						135			80 ⁹⁾	mA
Ios						-30 -112			4	mA
Tpd ²⁾						5			13	ns
Tpd ³⁾						6/ 7			30/ 30	ns
Tpd ⁴⁾						5			20	ns
Tpd ⁵⁾						6			25	ns
Tpd ⁶⁾						12			25	ns
Tpd ⁷⁾						8			27	ns
Tpd ⁸⁾						10			30	ns



1) 74AS881A

2) $C_n \rightarrow C_{n+4}$ 3) \bar{A} of $\bar{B} \rightarrow C_{n+4}$ { SUM mode
DIFF4) \bar{A} of $\bar{B} \rightarrow \bar{G}$, \bar{P} of \bar{F}_i (SUM en DIFF mode)5) \bar{A}_i of $\bar{B}_i \rightarrow \bar{F}_i$ (LOGIC mode)6) A of $B \rightarrow A=B$ (DIFF mode)7) \bar{A} of $\bar{B} \rightarrow \bar{P}$ (status check)8) \bar{A} of $\bar{B} \rightarrow C_{n+4}$ (status check)9) μA

De ALU kan actief-HOGE of actief-LAGE data
verwerken met de volgende pen-functies:

PIN NUMBER	2	1	23	22	21	20	19	18	9	10	11	13	7	16	15	17
Active-low data (Table I)	\bar{A}_0	\bar{B}_0	\bar{A}_1	\bar{B}_1	\bar{A}_2	\bar{B}_2	\bar{A}_3	\bar{B}_3	\bar{F}_0	\bar{F}_1	\bar{F}_2	\bar{F}_3	C_n	C_{n+4}	\bar{P}	\bar{G}
Active-high data (Table II)	A0	B0	A1	B1	A2	B2	A3	B3	F0	F1	F2	F3	Cn	Cn+4	X	Y

De ALU kan ook werken als comparator (ALU in
subtract-mode met $C_n = H$):

INPUT C_n	OUTPUT C_{n+4}	ACTIVE-LOW DATA (FIGURE 1)	ACTIVE-HIGH DATA (FIGURE 2)
H	H	$A \geq B$	$A \leq B$
H	L	$A < B$	$A > B$
L	H	$A > B$	$A < B$
L	L	$A \leq B$	$A \geq B$

Deel 6: Digitale rekenkundige schakelingen

In de logische mode kan de status van de ingangswaarden A en B en het uitgangswaard F worden gecheckt. De status-informatie op de \bar{P} , \bar{G} en C_{n+4} -uitgangen is dan gebaseerd op de logische combinaties:

$$\bar{P} = F_0 + F_1 + F_2 + F_3$$

$$\bar{G} = H$$

$$C_{n+4} = PC_n$$

FUNCTION TABLE FOR INPUT BITS EQUAL/NOT EQUAL

S0 = S3 = H, S1 = S2 = L, and M = H

C _n	DATA INPUTS				OUTPUTS		
	A0=B0	A1=B1	A2=B2	A3=B3	\bar{G}	\bar{P}	C _{n+4}
H	A0=B0	A1=B1	A2=B2	A3=B3	H	L	H
L	A0=B0	A1=B1	A2=B2	A3=B3	H	L	L
X	A0=B0	X	X	X	H	H	L
X	X	A1=B1	X	X	H	H	L
X	X	X	A2=B2	X	H	H	L
X	X	X	X	A3=B3	H	H	L

FUNCTION TABLE FOR INPUT PAIRS HIGH/NOT HIGH

S0 = S1 = S3 = L, S2 = H, and M = H

C _n	DATA INPUTS				OUTPUTS		
	A0 or B0=L	A1 or B1=L	A2 or B2=L	A3 or B3=L	\bar{G}	\bar{P}	C _{n+4}
H	A0 or B0=L	A1 or B1=L	A2 or B2=L	A3 or B3=L	H	L	H
L	A0 or B0=L	A1 or B1=L	A2 or B2=L	A3 or B3=L	H	L	L
X	A0=B0=H	X	X	X	H	H	L
X	X	A1=B1=H	X	X	H	H	L
X	X	X	A2=B2=H	X	H	H	L
X	X	X	X	A3=B3=H	H	H	L

Status-checks van ingangsparen (A_i, B_i):

S3	S2	S1	S0	M	$\bar{P} = F_0 + F_1 + F_2 + F_3$
L	H	L	L	H	$\bar{A}_0 \bar{B}_0 + \bar{A}_1 \bar{B}_1 + \bar{A}_2 \bar{B}_2 + \bar{A}_3 \bar{B}_3$
H	L	L	H	H	$(A_0 \oplus B_0) + (A_1 \oplus B_1) + (A_2 \oplus B_2) + (A_3 \oplus B_3)$

Toepassing 1 (aktief-LAAG)

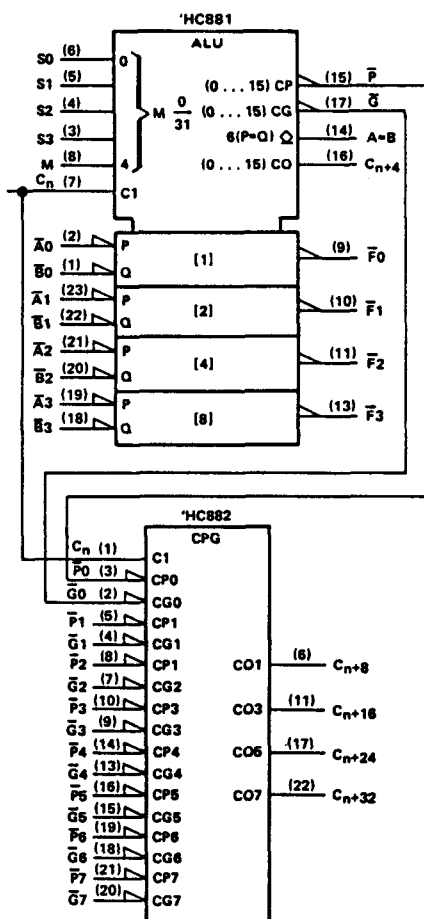
FIGURE 1
(USE WITH TABLE I)

TABLE I

SELECTION	S3 S2 S1 S0	M = H LOGIC FUNCTIONS	ACTIVE-LOW DATA	
			M = L: ARITHMETIC OPERATIONS	
			C _n = L (no carry)	C _n = H (with carry)
L L L L		$F = \bar{A}$	$F = A \text{ MINUS } 1$	$F = A$
L L L H		$F = \bar{A}\bar{B}$	$F = AB \text{ MINUS } 1$	$F = AB$
L L H L		$F = \bar{A} + B$	$F = \bar{A}\bar{B} \text{ MINUS } 1$	$F = \bar{A}\bar{B}$
L L H H		$F = 1$	$F = \text{MINUS } 1 \text{ (2's COMP)}$	$F = \text{ZERO}$
L H L L		$F = \bar{A} + \bar{B}$	$F = A \text{ PLUS } (A + \bar{B})$	$F = A \text{ PLUS } (A + \bar{B}) \text{ PLUS } 1$
L H L H		$F = B$	$F = AB \text{ PLUS } (A + \bar{B})$	$F = AB \text{ PLUS } (A + \bar{B}) \text{ PLUS } 1$
L H L L		$F = \bar{A} \oplus \bar{B}$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L H H L		$F = A + \bar{B}$	$F = A + \bar{B}$	$F = (A + \bar{B}) \text{ PLUS } 1$
H L L L		$F = \bar{A}\bar{B}$	$F = A \text{ PLUS } (A + B)$	$F = A \text{ PLUS } (A + B) \text{ PLUS } 1$
H L L H		$F = A \oplus B$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H L H L		$F = B$	$F = \bar{A}\bar{B} \text{ PLUS } (A + B)$	$F = \bar{A}\bar{B} \text{ PLUS } (A + B) \text{ PLUS } 1$
H L H H		$F = A + B$	$F = (A + B)$	$F = (A + B) \text{ PLUS } 1$
H H L L		$F = 0$	$F = A \text{ PLUS } A^*$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H H L H		$F = \bar{A}\bar{B}$	$F = AB \text{ PLUS } A$	$F = AB \text{ PLUS } A \text{ PLUS } 1$
H H H L		$F = AB$	$F = \bar{A}\bar{B} \text{ PLUS } A$	$F = \bar{A}\bar{B} \text{ PLUS } A \text{ PLUS } 1$
H H H H		$F = A$	$F = A$	$F = A \text{ PLUS } 1$

tabel 1

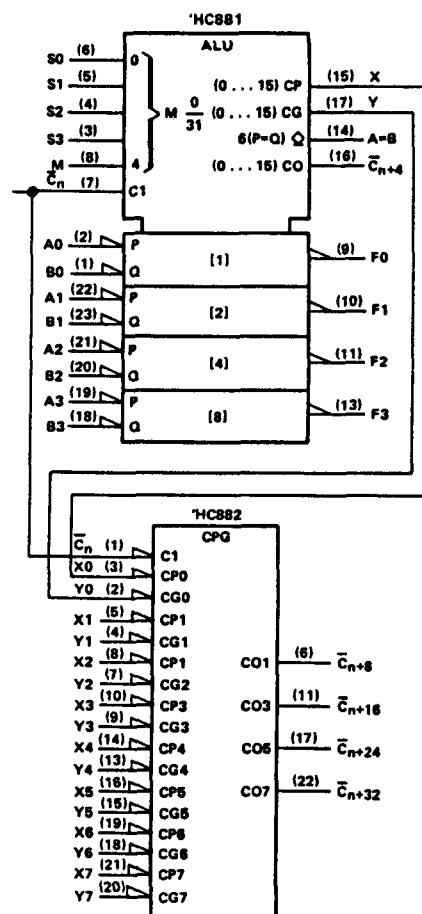
Toepassing 2 (aktief-HOOG)

TABLE II

SELECTION	S3 S2 S1 S0	M = H LOGIC FUNCTIONS	ACTIVE-HIGH DATA	
			M = L: ARITHMETIC OPERATIONS	
			C _n = H (no carry)	C _n = L (with carry)
L L L L		$F = \bar{A}$	$F = A$	$F = A \text{ PLUS } 1$
L L L H		$F = \bar{A} + B$	$F = A + B$	$F = (A + B) \text{ PLUS } 1$
L L H L		$F = \bar{A}\bar{B}$	$F = A + \bar{B}$	$F = (A + \bar{B}) \text{ PLUS } 1$
L L H H		$F = 0$	$F = \text{MINUS } 1 \text{ (2's COMP)}$	$F = \text{ZERO}$
L H L L		$F = \bar{A}\bar{B}$	$F = A \text{ PLUS } \bar{A}\bar{B}$	$F = A \text{ PLUS } \bar{A}\bar{B} \text{ PLUS } 1$
L H L H		$F = \bar{B}$	$F = (A + B) \text{ PLUS } \bar{A}\bar{B}$	$F = (A + B) \text{ PLUS } \bar{A}\bar{B} \text{ PLUS } 1$
L H L H		$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L H H L		$F = \bar{A}\bar{B} \text{ MINUS } 1$	$F = \bar{A}\bar{B}$	$F = \bar{A}\bar{B}$
H L L L		$F = \bar{A} + B$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB \text{ PLUS } 1$
H L L H		$F = \bar{A} \oplus \bar{B}$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H L H L		$F = B$	$F = (A + \bar{B}) \text{ PLUS } AB$	$F = (A + \bar{B}) \text{ PLUS } AB \text{ PLUS } 1$
H L H H		$F = AB$	$F = AB \text{ MINUS } 1$	$F = AB$
H H L L		$F = 1$	$F = A \text{ PLUS } A^*$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H H L H		$F = A + \bar{B}$	$F = (A + B) \text{ PLUS } A$	$F = (A + B) \text{ PLUS } A \text{ PLUS } 1$
H H H L		$F = A + B$	$F = (A + \bar{B}) \text{ PLUS } A$	$F = (A + \bar{B}) \text{ PLUS } A \text{ PLUS } 1$
H H H H		$F = A$	$F = A \text{ MINUS } 1$	$F = A$

*Each bit is shifted to the next more significant position.

tabel 2

FIGURE 2
(USE WITH TABLE II)

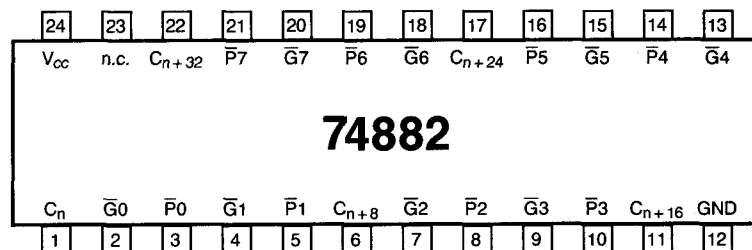
6.2 74xx-serie TTL en HC

74882

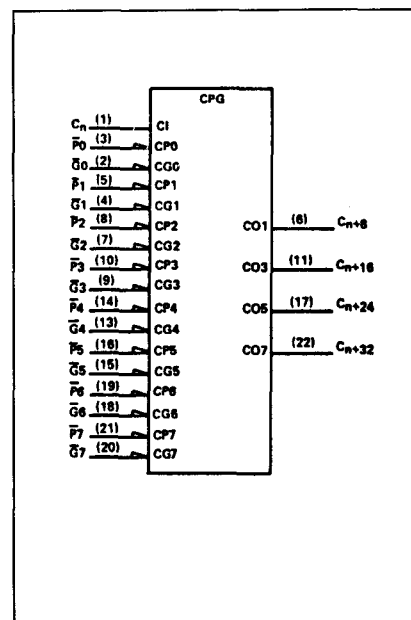
32-bit Look-ahead Carry generator

(carry-functies compatibel met ALU)

Figuur 6/6.2-882



LOGICA	TTL	L	F	S	LS	AS	ALS	C	HC	
	VARIABELE PARAMETERS									Eenheid
Icc						72			80 ⁶⁾	mA
Ios						-150			4	mA
Tpd ¹⁾						4 14			—	ns
Tpd ²⁾						2 8			—	ns
Tpd ³⁾						2 8			—	ns
Tpd ⁴⁾						2 10			—	ns
Tpd ⁵⁾						2 12			—	ns



1) Cn → elke uitgang

2) P of G → Cn+8

3) P of G → Cn+16

4) P of G → Cn+24

5) P of G → Cn+32

6) μA

$$C_{n+8} = G_1 + P_1 G_0 + P_1 P_0 C_n$$

$$C_{n+16} = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_n$$

$$C_{n+24} = G_5 + P_5 G_4 + P_5 P_4 G_3 + P_5 P_4 P_3 G_2 + P_5 P_4 P_3 P_2 G_1 + P_5 P_4 P_3 P_2 P_1 G_0 + P_5 P_4 P_3 P_2 P_1 P_0 C_n$$

$$C_{n+32} = G_7 + P_7 G_6 + P_7 P_6 G_5 + P_7 P_6 P_5 G_4 + P_7 P_6 P_5 P_4 G_3 + P_7 P_6 P_5 P_4 P_3 G_2 + P_7 P_6 P_5 P_4 P_3 P_2 G_1 + P_7 P_6 P_5 P_4 P_3 P_2 P_1 G_0 + P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0 C_n$$

logische vergelijkingen
voor de 74882

6.2 74xx-serie TTL en HC

waarheidstabellen

FUNCTION TABLE
FOR C_n+32 OUTPUT

INPUTS																	OUTPUT
\bar{G}_7	\bar{G}_6	\bar{G}_5	\bar{G}_4	\bar{G}_3	\bar{G}_2	\bar{G}_1	\bar{G}_0	\bar{P}_7	\bar{P}_6	\bar{P}_5	\bar{P}_4	\bar{P}_3	\bar{P}_2	\bar{P}_1	\bar{P}_0	C_n	C_n+32
L	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	H
X	L	X	X	X	X	X	X	L	X	X	X	X	X	X	X	X	H
X	X	L	X	X	X	X	X	L	L	X	X	X	X	X	X	X	H
X	X	X	L	X	X	X	X	L	L	L	X	X	X	X	X	X	H
X	X	X	X	L	X	X	X	L	L	L	L	X	X	X	X	X	H
X	X	X	X	X	L	X	X	L	L	L	L	L	X	X	X	X	H
X	X	X	X	X	X	L	X	L	L	L	L	L	L	X	X	X	H
X	X	X	X	X	X	X	L	L	L	L	L	L	L	L	X	X	H
X	X	X	X	X	X	X	X	L	L	L	L	L	L	L	L	H	H
All other combinations																	L

FUNCTION TABLE
FOR C_n+24 OUTPUT

INPUTS													OUTPUT
G5	G4	G3	G2	G1	G0	P5	P4	P3	P2	P1	P0	Cn	Cn+24
L	X	X	X	X	X	X	X	X	X	X	X	X	H
X	L	X	X	X	X	L	X	X	X	X	X	X	H
X	X	L	X	X	X	L	L	X	X	X	X	X	H
X	X	X	L	X	X	L	L	L	X	X	X	X	H
X	X	X	X	L	X	L	L	L	L	X	X	X	H
X	X	X	X	X	L	L	L	L	L	L	X	X	H
X	X	X	X	X	X	L	L	L	L	L	L	H	H
All other combinations													L

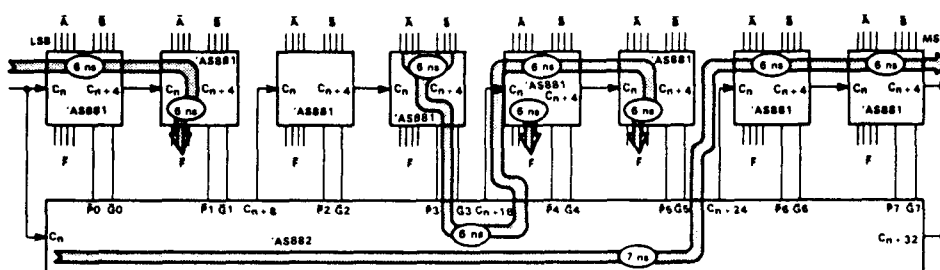
FUNCTION TABLE
FOR C_n+16 OUTPUT

INPUTS									OUTPUT
\bar{G}_3	\bar{G}_2	\bar{G}_1	\bar{G}_0	\bar{P}_3	\bar{P}_2	\bar{P}_1	\bar{P}_0	C_n	C_n+16
L	X	X	X	X	X	X	X	X	H
X	L	X	X	L	X	X	X	X	H
X	X	L	X	L	L	X	X	X	H
X	X	X	L	L	L	X	X	X	H
X	X	X	X	L	L	L	L	H	L
All other combinations									L

FUNCTION TABLE
FOR C_n+8 OUTPUT

INPUTS					OUTPUT
\bar{G}_1	\bar{G}_0	\bar{P}_1	\bar{P}_0	C_n	$C_n + 8$
L	X	X	X	X	H
X	L	L	X	X	H
X	X	L	L	H	H
All other combinations					L

Any inputs not shown in a given table are irrelevant with respect to that output.



Toepassingsvoorbeeld: look-ahead carry voor een 32-bits ALU (hier de 74AS881). De vermelde vertragingstijden gelden bij de 74AS882 (met $C_L = 15\text{pF}$).

6.2 74xx-serie TTL en HC

6/7

Diversen

Inhoud

6/7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL (aanvulling 62)

8260	arithmetic logic element (ALU)
8261	fast carry expander
8262	8 bit parity generator/checker
8268	gated full adder
8269	4 bit comparator
8282	4 bit BCD arithmetic unit
8283	4 bit BCD adder

6/7.3 Diverse schakelingen uit de FC-serie DTL (aanvulling 62)

FCH 281	5 bit comparator
FCH 291	10 bit parity checker

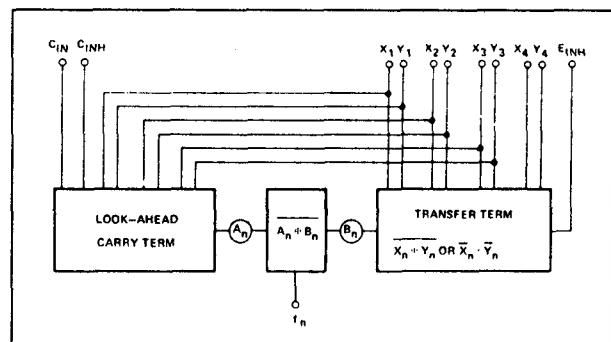
6/7.2

Diverse schakelingen
uit de 8xxx-serie DCL en TTL

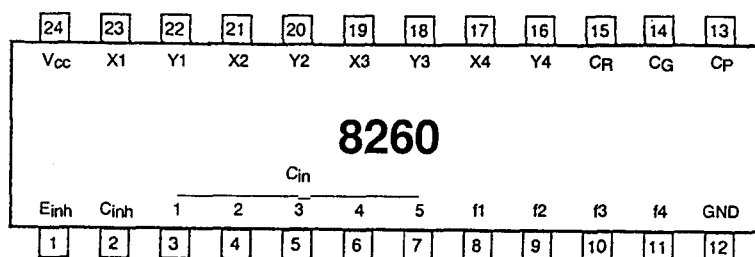
8260

Arithmetic Logic Element

Dit is een monolithisch gate-array met vier full-adders, gestructureerd in een look-ahead mode die - door een juiste adres-sering van de inhibit-lijnen - gebruikt kan worden als vier onafhankelijke exclusive NOR- en NAND-poorten. Bij toepassing als 4 bit adder kunnen vier sets data snel bij elkaar worden opgeteld. Wanneer "true" ingangsvariabelen worden gebruikt, verschijnt de "true" som op de f-uitgang; geïnverteerde ingangsvariabelen produceren het comple-ment van de som van de true variabelen. De volgende carry-out's staan ter beschikking: intern gegenereerde (CG), propagated (CP) en ripple (CR).

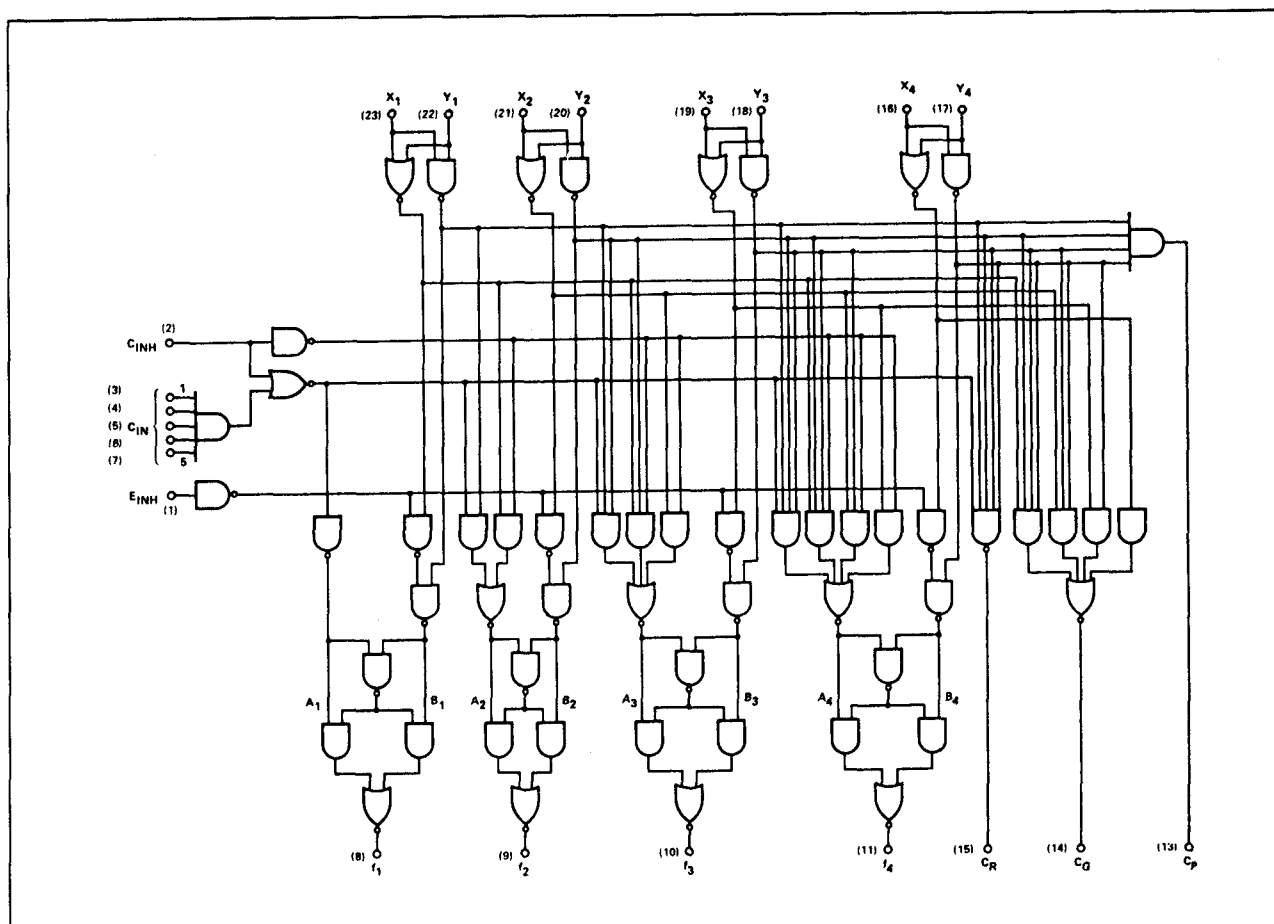


Blokschema.



Aansluitgegevens.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL



Logisch schema.

$C_{INH} = 1 \longrightarrow A_n = 1$ $C_{INH} = 0 \longrightarrow A_n = \downarrow$													
C_{IN}	A_1	A_1	X_1	Y_1	A_2	A_2	X_2	Y_2	A_3	A_3	X_3	Y_3	A_4
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	1	0	0	0	1	0
		0	1	0	0	0	1	0	0	0	1	0	0
		0	1	1	1	0	1	1	1	0	1	1	1
		1	0	0	0	1	0	0	0	1	0	0	0
	1	0	1	1	1	0	1	1	1	0	1	1	
	1	1	0	1	1	1	0	1	1	1	0	1	
	1	1	1	1	1	1	1	1	1	1	1	1	

A_n	B_n	f_n
0	0	1
0	1	0
1	0	0
1	1	1

E_{INH}	X_n	Y_n	B_n
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

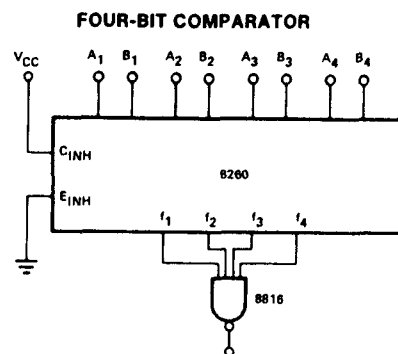
Waarheidstabel.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

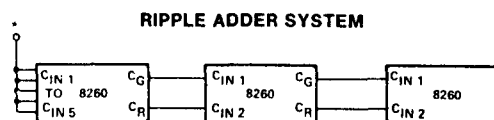
INPUTS	Least Significant C _{IN} Inputs to be*	CONTROLS		f	
		C _{INH}	E _{INH}		
X_n, Y_n	0	0	0	Σ_n	Add
	0	0	1	--	Not used
	0	1	0	$X_n Y_n + \bar{X}_n \bar{Y}_n$	Coincidence
	0	1	1	$X_n Y_n$	AND
\bar{X}_n, \bar{Y}_n	1	0	0	Σ_n	Add
	1	0	1	---	Not Used
	1	1	0	$\bar{X}_n \bar{Y}_n + X_n Y_n$	Coincidence
	1	1	1	$\bar{X}_n \bar{Y}_n$	AND

*Least significant of a "Multiple Package"-adder system.

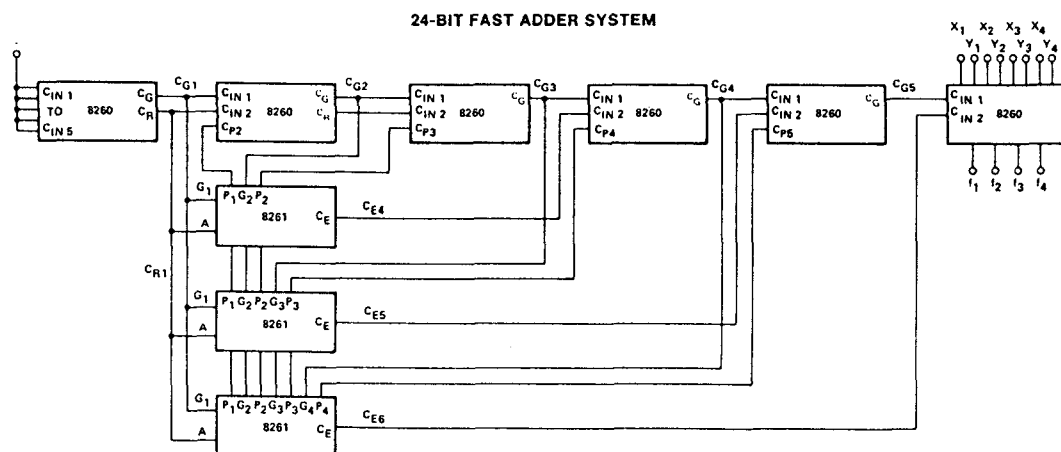
Instelling bedrijfsmodes.



Toepassingsvoorbeeld 2.

*Tied to V_{CC} if not-true inputs are used, otherwise to ground.

Toepassingsvoorbeeld 1.

*Tied to V_{CC} if not-true inputs are used, otherwise to ground.

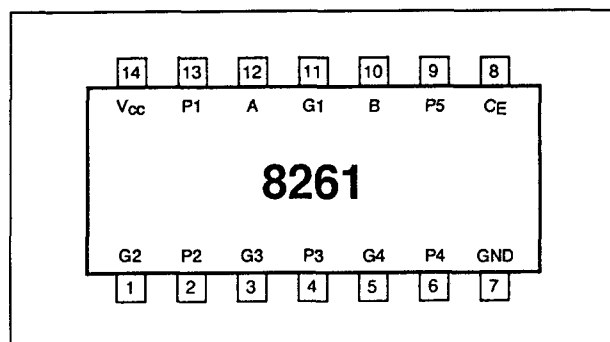
Toepassingsvoorbeeld 3.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

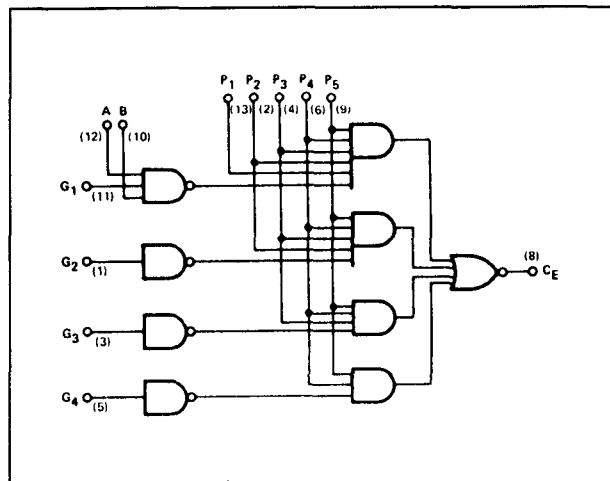
8261

Fast Carry Expander

Dit is een monolithisch gate-array, dat speciaal werd ontwikkeld om samen te werken met de 8260 ALU. De 8261/8260 combinatie vereenvoudigt de implementatie van de look-ahead techniek in optel-systemen.



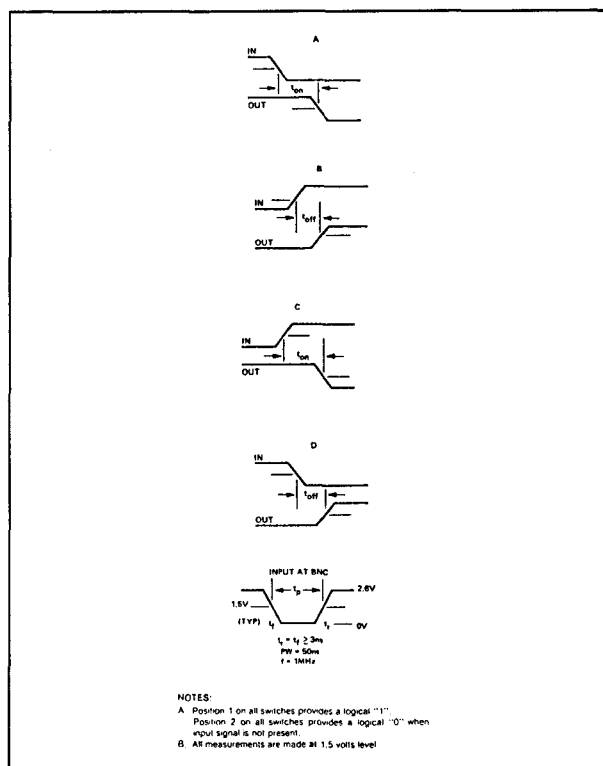
Aansluitgegevens.



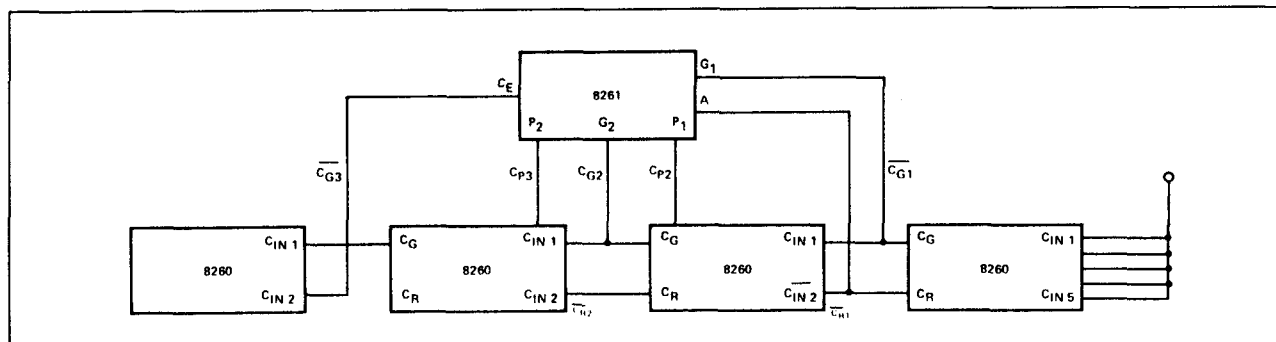
Logisch schema.

PIN DESIGNATION	INPUT							WAVEFORM
	A	B	G ₁	G ₂	G ₃	G ₄	P ₄	
1	PULSE	1	1	1	1	1	1	A,B
2	1	PULSE	1	1	1	1	1	
3	1	1	PULSE	1	1	1	1	
4	1	1	1	PULSE	1	1	1	
5	1	1	1	1	PULSE	1	1	
6	1	1	1	1	1	PULSE	1	
7	2	2	2	2	2	2	PULSE	C,D

Testtabel.



De bij de testtabel behorende golfvormen.



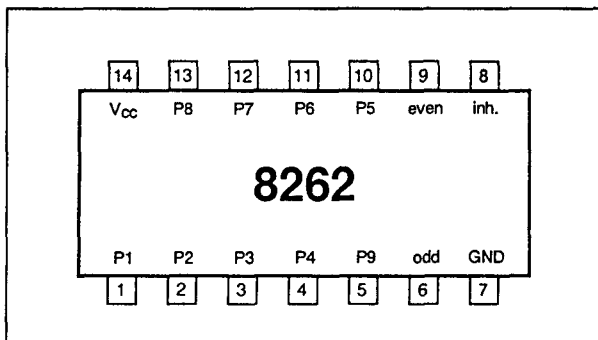
Toepassingsvoorbeeld.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

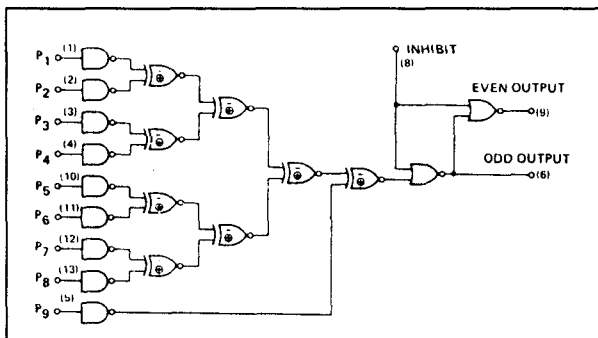
8262

8 bit Parity Generator/Checker

Deze pariteitsgenerator/checker heeft 9 ingangen en wordt vaak gebruikt om fouten te detecteren bij data-transmissie of bij het terugwinnen van data. Er zijn twee uitgangen (ODD en EVEN), die met een inhibit-sigitaal kunnen worden uitgeschakeld (met inhibit = "1" worden beide uitgangen "0"). Bij gebruik als pariteitsgenerator voegt de 8262 een pariteitsbit toe aan de data. Aan de ontvangzijde wordt de juiste pariteit gecheckt.



Aansluitgegevens.



Logisch schema.

$$\begin{aligned} \text{Odd} &= P_1 \oplus P_2 \oplus P_3 \oplus P_4 \oplus P_5 \oplus P_6 \oplus P_7 \oplus P_8 \oplus P_9 \\ \text{Even} &= P_1 \oplus P_2 \oplus P_3 \oplus P_4 \oplus P_5 \oplus P_6 \oplus P_7 \oplus P_8 \oplus P_9 \end{aligned}$$

Positieve logika.

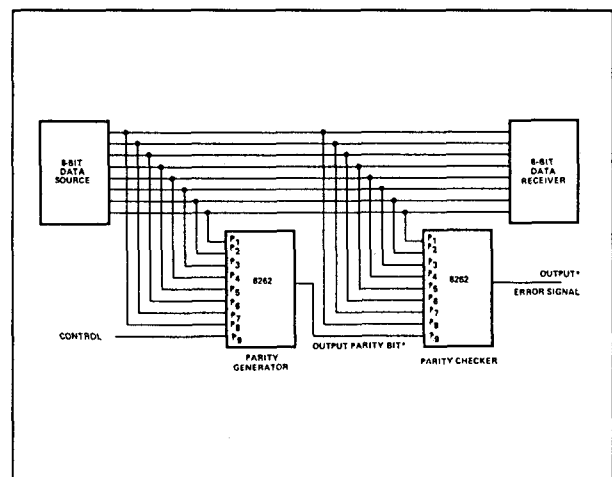
MEASURE DELAY FROM	SWITCH POSITION			WAVEFORM	
	INH	P ₈	P ₉	EVEN	ODD
P ₈ to ODD	1	2	1		1
P ₉ to ODD	1	1	2		2
P ₈ to EVEN	1	2	1	2	
P ₉ to EVEN	1	1	2	1	
INH to EVEN	2	1	1	2	

Waarheidstabel.

Test No.	Inputs										Outputs	
	P1	P2	P3	P4	P5	P6	P7	P8	P9	INH	Even	Odd
1	PG	0	0	0	0	0	0	0	0	0	T	T
2	0	0	PG	0	0	0	0	0	0	0	T	T
3	0	0	0	0	PG	0	0	0	0	0	T	T
4	0	0	0	0	0	0	PG	0	0	0	T	T
5	0	0	0	0	0	0	0	0	PG	0	T	T
6	0	0	0	0	0	0	0	0	0	PG	T	
7	0	0	0	0	0	0	0	0	1	PG		T

"1" = 2.7V "0" = Ground "T" = Test

Testtabel.



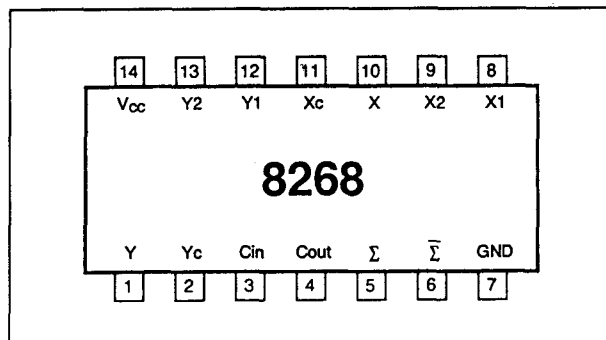
Toepassingsvoorbeeld.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

8268

Gated Full Adder

Dit is een enkel bit volledige opteller (full adder) met afschakelbare "true" en "complementaire" ingangen, complementaire som (Σ en $\bar{\Sigma}$) uitgangen en een geïnverteerde carry-uitgang.



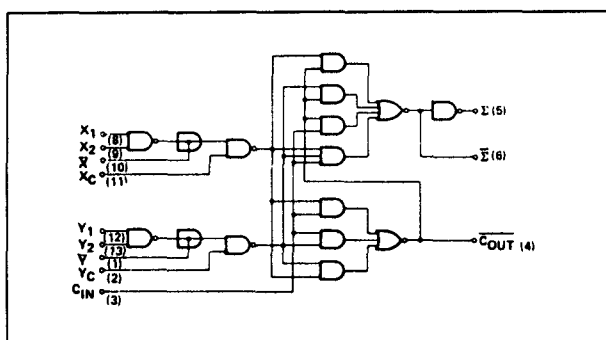
Aansluitgegevens.

C _{IN}	Y	X	C _{OUT}	Σ	Σ̄
0	0	0	1	0	1
0	0	1	1	1	0
0	1	0	1	1	0
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	1	0

NOTES:

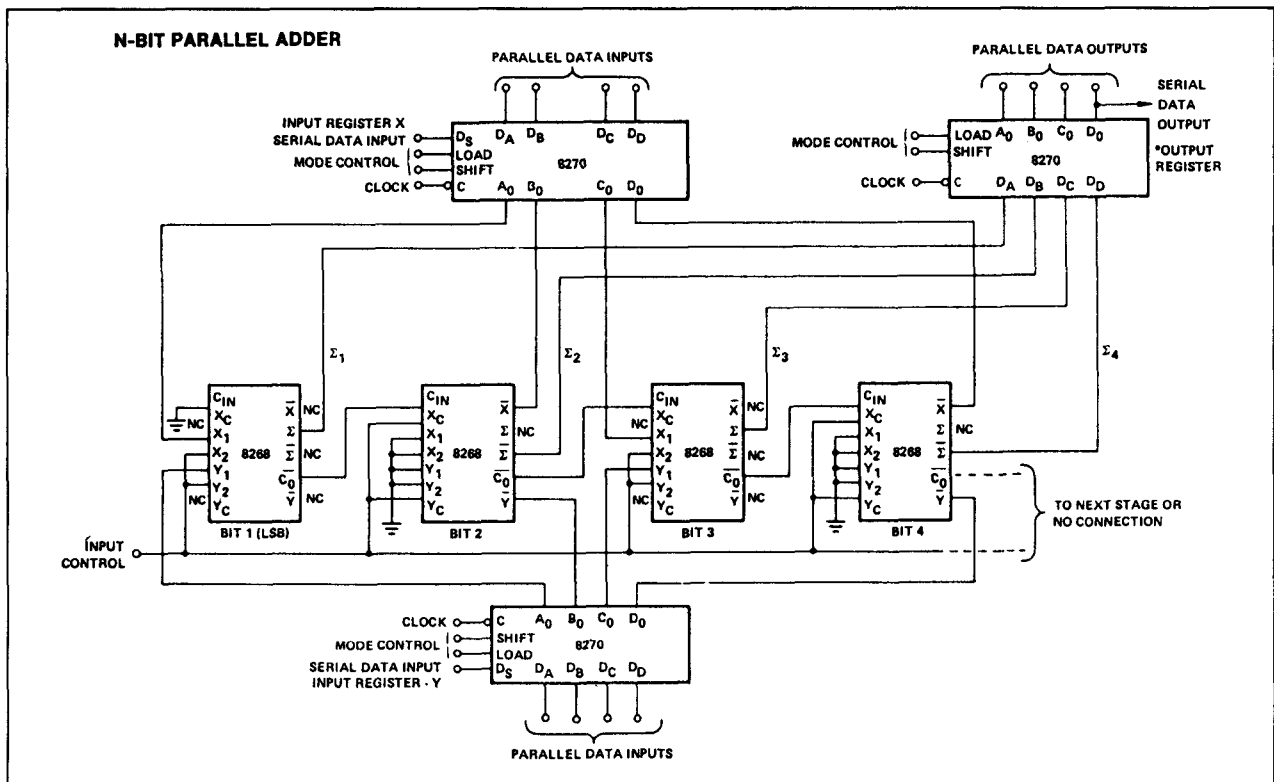
1. $X = \bar{X} + X_c$, $Y = \bar{Y} + Y_c$, where $\bar{X} = \bar{X}_1 + \bar{X}_2$, $\bar{Y} = \bar{Y}_1 + \bar{Y}_2$
2. When \bar{X} or \bar{Y} are used as inputs, X_1 and X_2 or Y_1 and Y_2 respectively must be tied to GND.
3. When X_1 and X_2 or Y_1 and Y_2 are used as inputs, \bar{X} or \bar{Y} respectively must be left open or used to perform the WIRED-AND function.

Waarheidstabel.

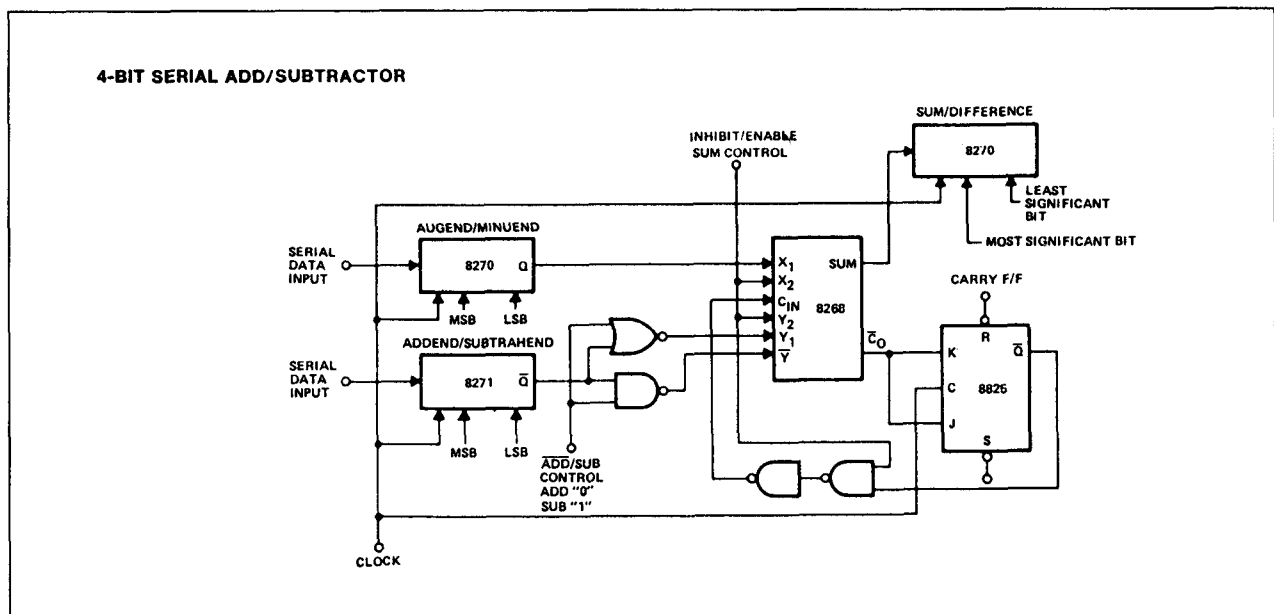


Logisch schema.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL



Toepassingsvoorbeeld 1.

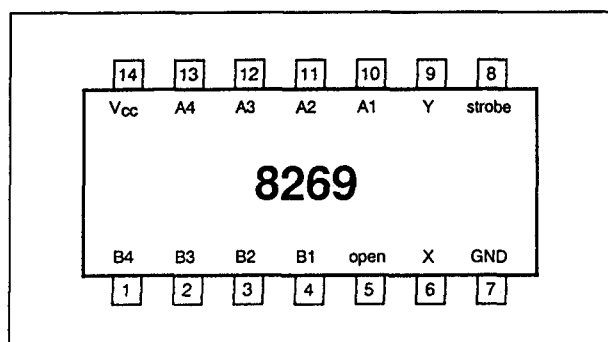


Toepassingsvoorbeeld 2.

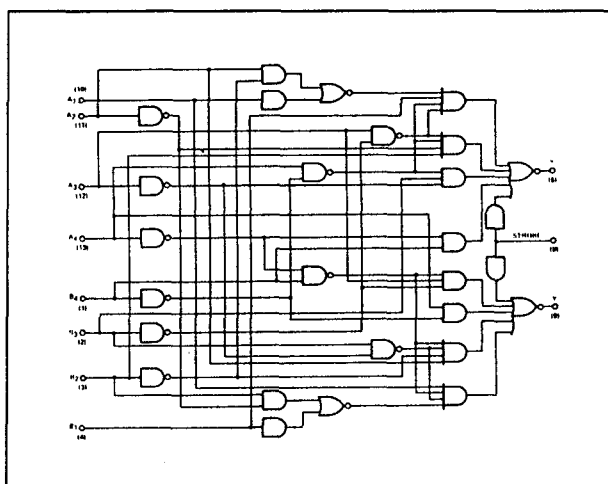
7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

8269**4 bit Comparator**

Dit is een array van poorten, die zijn ingericht om twee 4 bit getallen numeriek te vergelijken. De uitgangen geven aan of de beide getallen gelijk zijn of welk getal groter is.



Aansluitgegevens.



Logisch schema.

INPUT			OUTPUT	
A_n	B_n	STROBE	X	Y
A	>	B	0	1
A	<	B	0	0
A	=	B	0	1
A	≥	B	1	0

Waarheidstabel.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

8282

4 bit BCD Arithmetic Unit

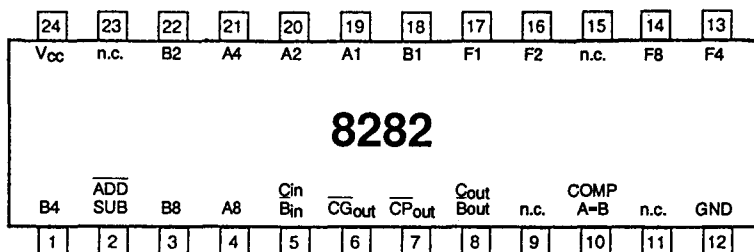
Dit is een snelle binair gecodeerde (BCD) rekenkundige schakeling met look-ahead carry/borrow. Afhankelijk van de toestand van de $\overline{\text{ADD/SUB}}$ besturingslijn produceert de schakeling de BCD som of verschil van twee decimale getallen die op de BCD-ingangen worden aangeboden (in "gewogen" BCD-formaat). Ook is er een vergelijkings-uitgang ($A = B$): in de aftrek-mode geeft deze uitgang aan of beide getallen gelijk zijn.

BCD CODE				
DECIMAL EQUIVALENT	BCD NUMBER			
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Waarheidstabel.

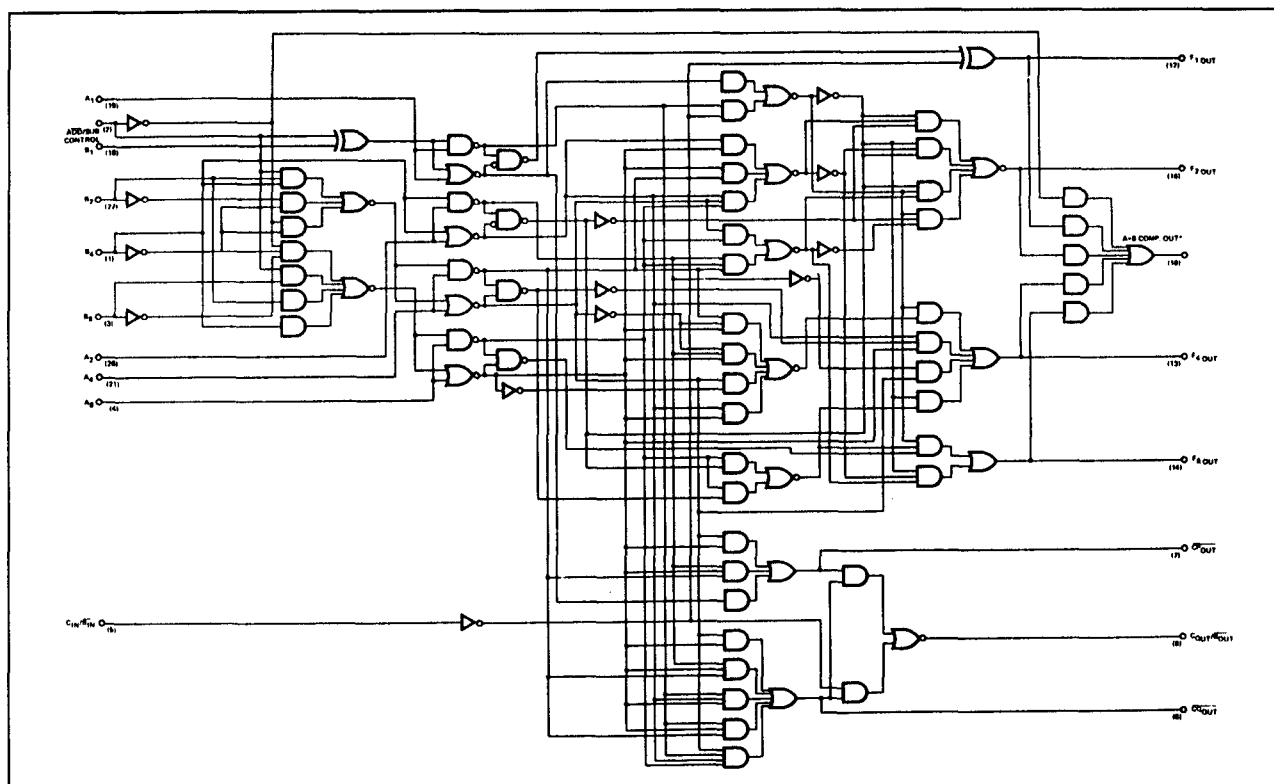
FUNCTION	$\overline{\text{ADD/SUB}}$	A(A ₈ ,A ₄ ,A ₂ ,A ₁)	B(B ₈ ,B ₄ ,B ₂ ,B ₁)	C _{in} /B _{in}	F(F ₈ ,F ₄ ,F ₂ ,F ₁)	C _{out} /B _{out}	COMPARE (A = B)
Add	0	BCD Augend	BCD Addend	1=Carry 0=No Carry	IF C _{in} = 1 F = A + B + 1 IF C _{in} = 0 F = A + B	F < B C _{out} /B _{out} = 0 F > B C _{out} /B _{out} = 1	X
Subtract	1	BCD Minuend	BCD Subtrahend	0=Borrow 1=No Borrow	IF B _{in} = 0 F = A - B - 1 IF B _{in} = 1 F = A - B	A > B C _{out} /B _{out} = 1 A < B C _{out} /B _{out} = 0 A < B C _{out} /B _{out} = 0 A > B C _{out} /B _{out} = 1	X
Compare	1	BCD Word A	BCD Word B	1	A - B	A < B C _{out} /B _{out} = 0 A > B C _{out} /B _{out} = 1	IF A = B Compare = 1 IF A > B Compare = 0
Binary to BCD Conversion	0	0 ≤ A < 15	B = 0	X	BCD	A < 9 C _{out} = 0 A ≥ 9 C _{out} = 1	X

Functietabel.

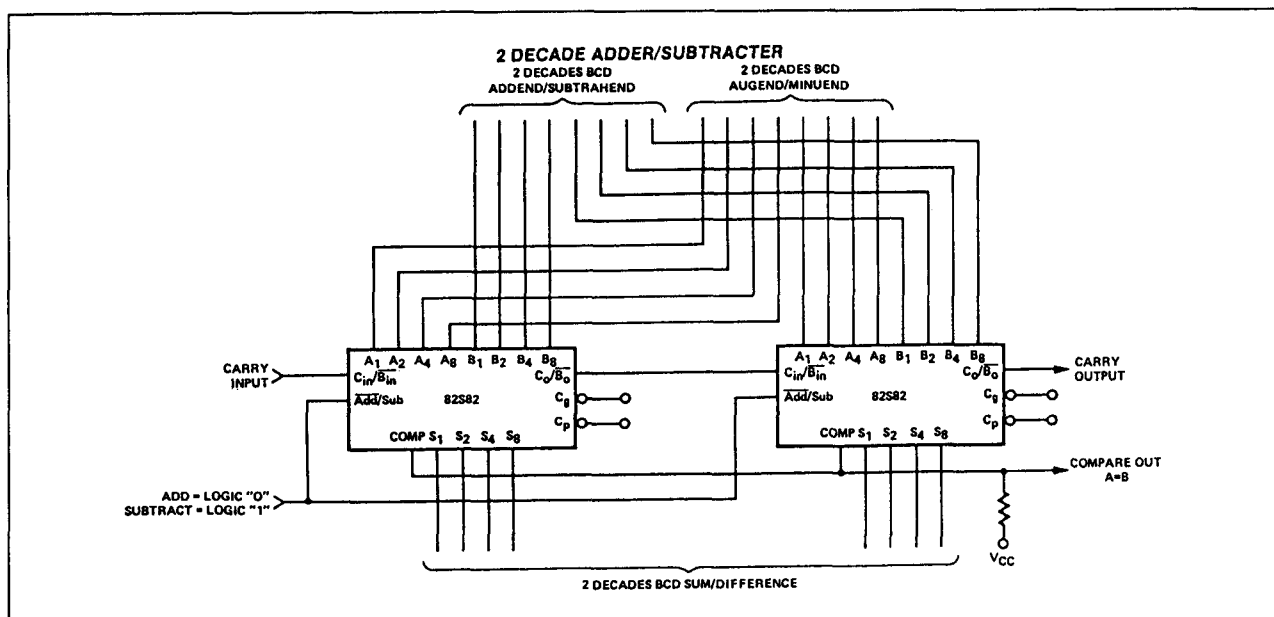


Aansluitgegevens.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

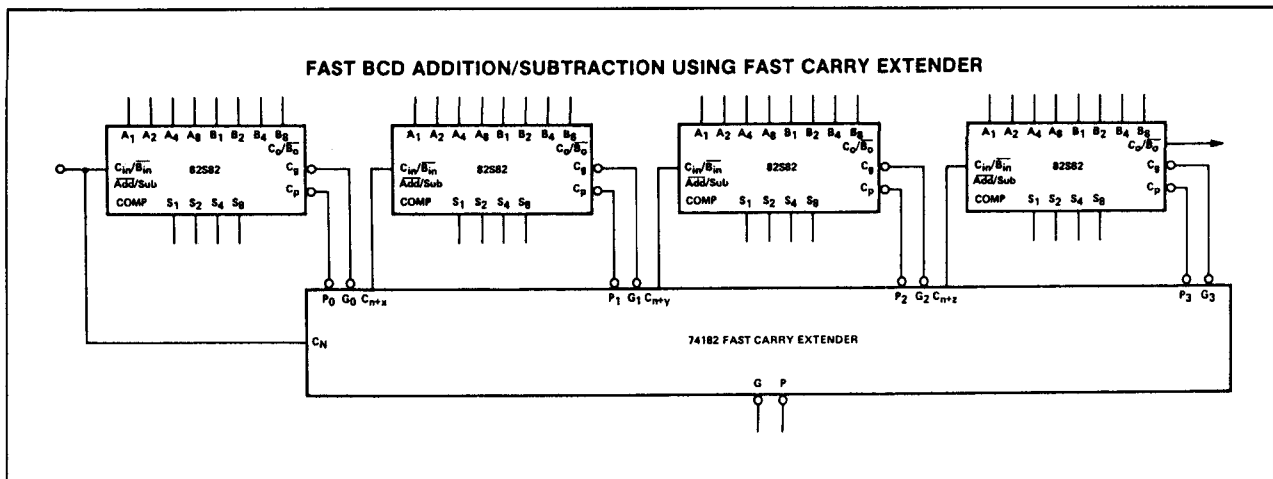


Logisch schema.

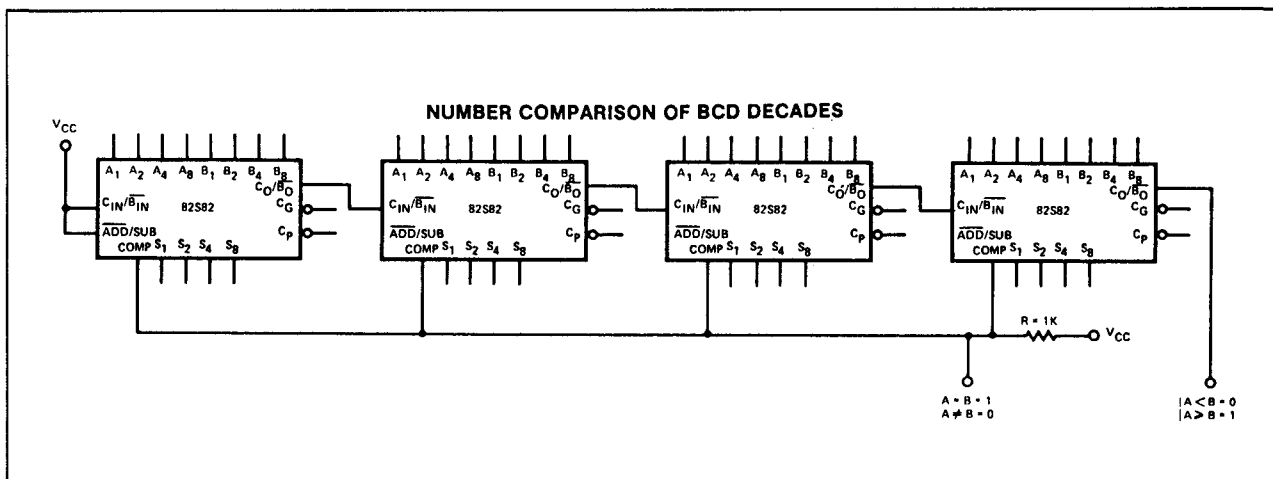


Toepassingsvoorbeeld 1.

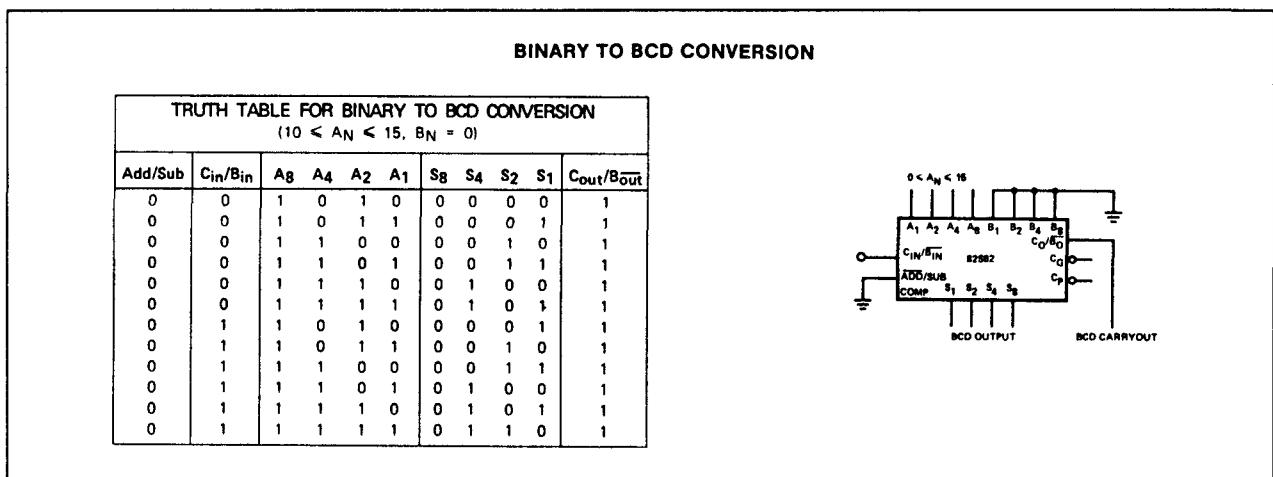
7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL



Toepassingsvoorbeeld 2.

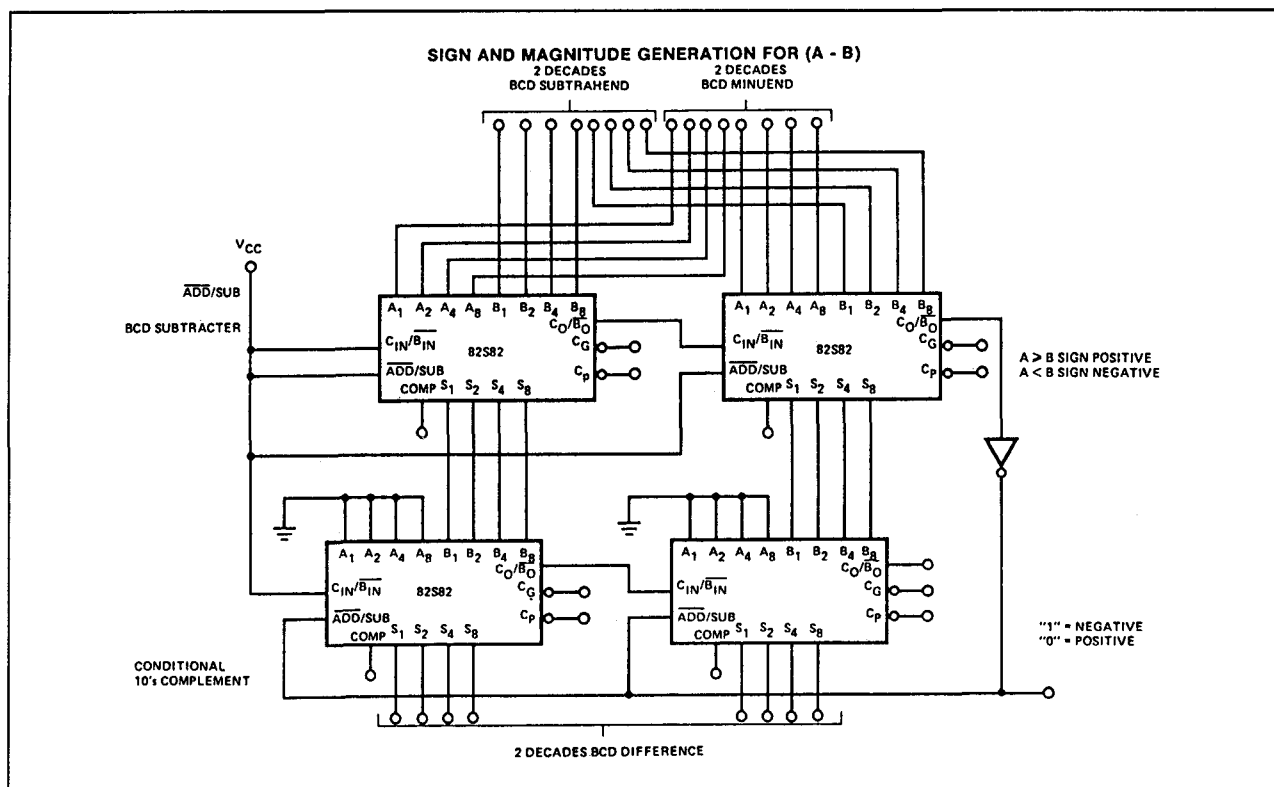


Toepassingsvoorbeeld 3.



Toepassingsvoorbeeld 4.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

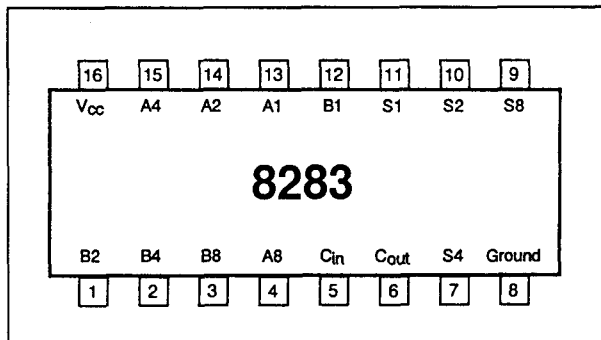


Toepassingsvoorbeeld 5.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

8283**4 bit BCD Adder**

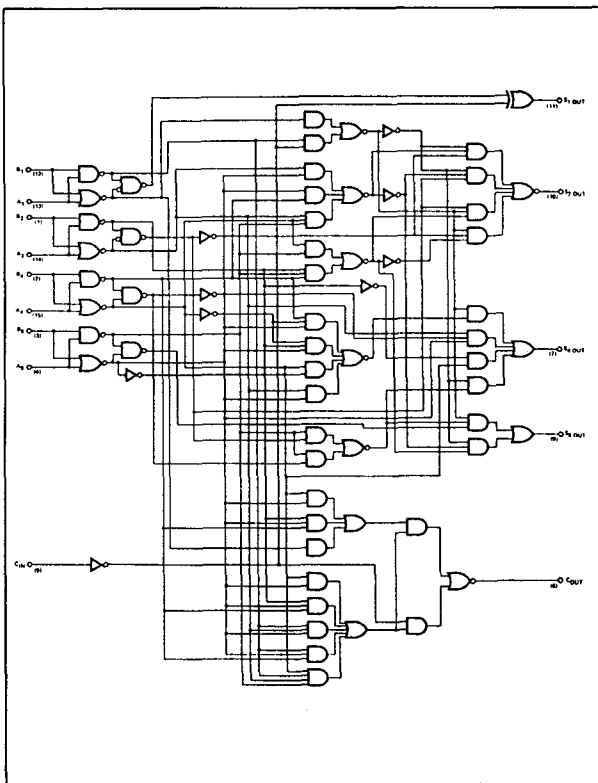
Deze schakeling produceert de binair geco-deerde (BCD) som van twee decimale ge-tallen die in gewogen 8-4-2-1 formaat op de ingangen staan. De opteller is voorzien van carry-in en carry-uit aansluitingen (voor gemakkelijke uitbreiding).



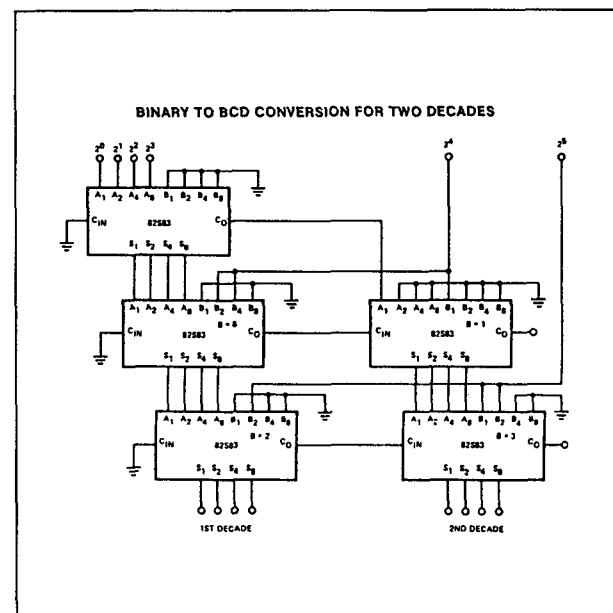
Aansluitgegevens.

BCD CODE				
Decimal Equivalent	BCD Number			
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Waarheidstabel.

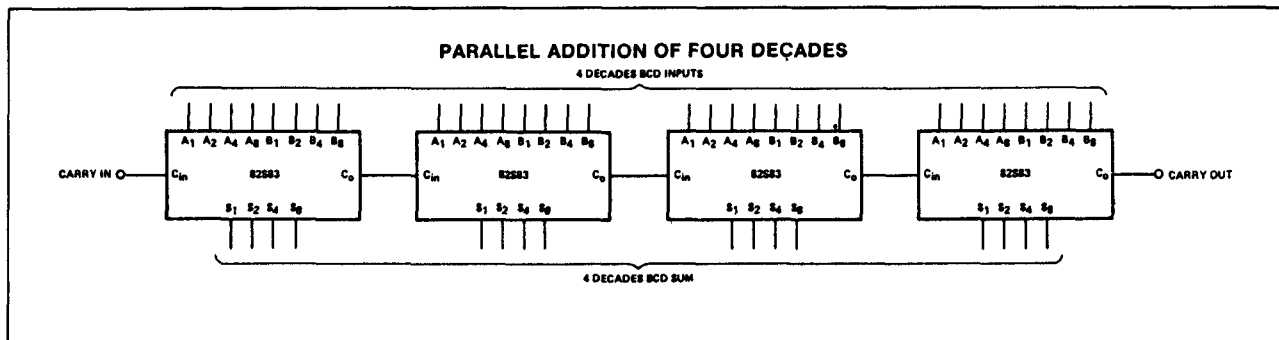


Logisch schema.

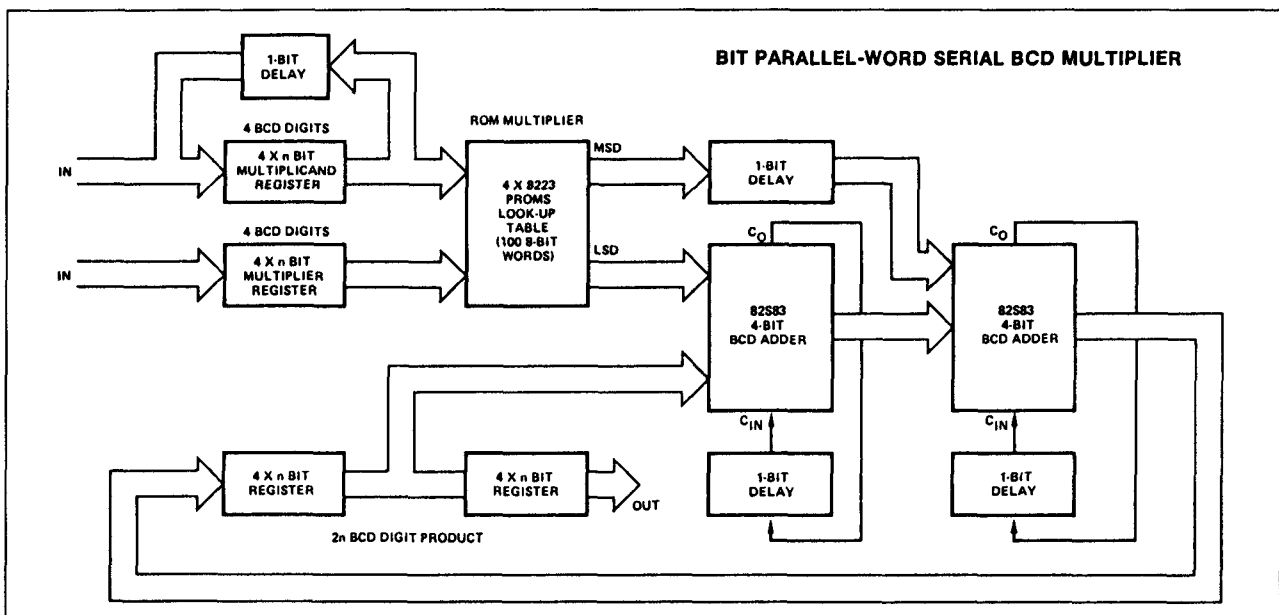


Toepassingsvoorbeeld 1.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL



Toepassingsvoorbeeld 2.



Toepassingsvoorbeeld 3.

BINARY TO BCD CONVERSION USING A_1 INPUTS

PARTIAL TRUTH TABLE FOR $A_1 > 9, B_1 = 0$

CIN	A1	A2	A4	A8	B1	B2	B4	B8	S1	S2	S4	S8	CO
0	0	1	0	1	0	0	0	0	0	0	0	0	1
0	1	1	0	1	0	0	0	0	1	0	0	0	1
0	0	0	1	1	0	0	0	0	0	1	0	0	1
0	1	0	1	1	0	0	0	0	1	1	0	0	1
0	0	1	1	1	0	0	0	0	0	0	1	0	1
0	1	1	1	1	0	0	0	0	1	0	1	0	1
1	0	1	0	1	0	0	0	0	1	0	0	0	1
1	1	1	0	1	0	0	0	0	0	1	0	0	1
1	0	0	1	1	0	0	0	0	1	1	0	0	1
1	1	0	1	1	0	0	0	0	0	0	1	0	1
1	0	1	1	1	0	0	0	0	1	0	1	0	1
1	1	1	1	1	0	0	0	0	0	1	1	0	1

LEAST SIGNIFICANT DECADE

Bijbehorende waarheidstabel 1.

7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

BINARY TO BCD CONVERSION USING B₁ INPUTS

PARTIAL TRUTH TABLE FOR B₁ > 9, A₁ = 0

CIN	A1	A2	A4	A8	B1	B2	B4	B8	S1	S2	S4	S8	CO
0	0	0	0	0	0	1	0	1	0	0	0	0	1
0	0	0	0	0	1	1	0	1	1	0	0	0	1
0	0	0	0	0	0	0	1	1	0	1	0	0	1
0	0	0	0	0	1	0	1	1	1	1	0	0	1
0	0	0	0	0	0	1	1	1	1	0	0	1	0
0	0	0	0	0	1	1	1	1	1	0	1	0	1
1	0	0	0	0	0	1	0	1	1	0	0	0	1
1	0	0	0	0	1	1	0	1	0	1	0	0	1
1	0	0	0	0	0	0	1	1	1	1	0	0	1
1	0	0	0	0	1	0	1	1	0	0	1	0	1
1	0	0	0	0	0	1	1	1	1	0	1	0	1
1	0	0	0	0	1	0	0	0	0	1	1	0	1

Bijbehorende waarheidstabel 2.

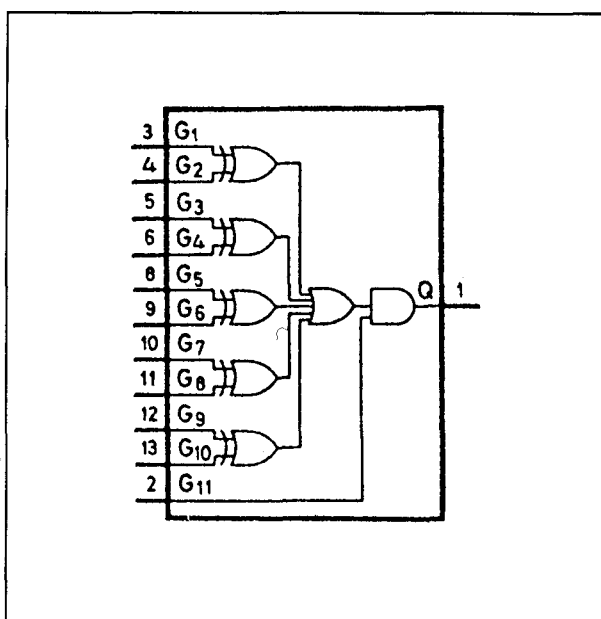
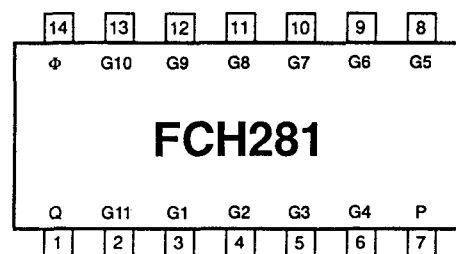
7.2 Diverse schakelingen uit de 8xxx-serie DCL en TTL

6/7.3

Diverse schakelingen
uit FC-serie DTL**FCH 281****enkele 5 bit
comparator**

Wanneer bij één of meer paren (G1-G2, etc, G9-G10) één ingang LAAG is en de andere HOOG zal de uitgang HOOG zijn (als G11 tenminste ook HOOG is). In de andere gevallen zal de uitgang LAAG zijn.
(fan-out = 8)

Figuur 6/7.3-281.



Logisch symbol.

7.3 Diverse schakelingen uit de FC-serie DTL

$G_1 G_2$	$G_3 G_4$	$G_5 G_6$	$G_7 G_8$	$G_9 G_{10}$	G_{11}	Q
Equal	Equal	Equal	Equal	Equal	H	L
Unequal	X	X	X	X	H	H
X	Unequal	X	X	X	H	H
X	X	Unequal	X	X	H	H
X	X	X	Unequal	X	H	H
X	X	X	X	Unequal	H	H
X	X	X	X	X	L	L

G_1	G_2	
L	L	Equal
L	H	Unequal
H	L	Unequal
H	H	Equal

H = HIGH state (the more positive voltage)

L = LOW state (the less positive voltage)

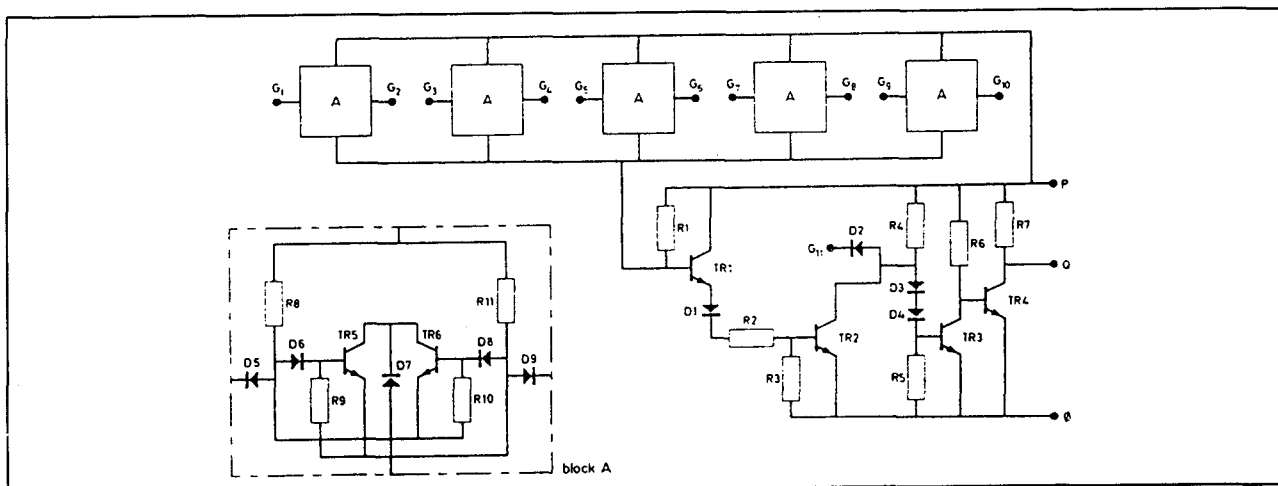
X = state is immaterial

Waarheidstabellen.

LOGIC FUNCTION

$$Q = \left[(\overline{G_1} \cdot \overline{G_2} + G_1 \cdot G_2) + (\overline{G_3} \cdot \overline{G_4} + G_3 \cdot G_4) + (\overline{G_5} \cdot \overline{G_6} + G_5 \cdot G_6) + (\overline{G_7} \cdot \overline{G_8} + G_7 \cdot G_8) + (\overline{G_9} \cdot \overline{G_{10}} + G_9 \cdot G_{10}) \right] \cdot G_{11}$$

Positieve logica.



Functioneel schema.

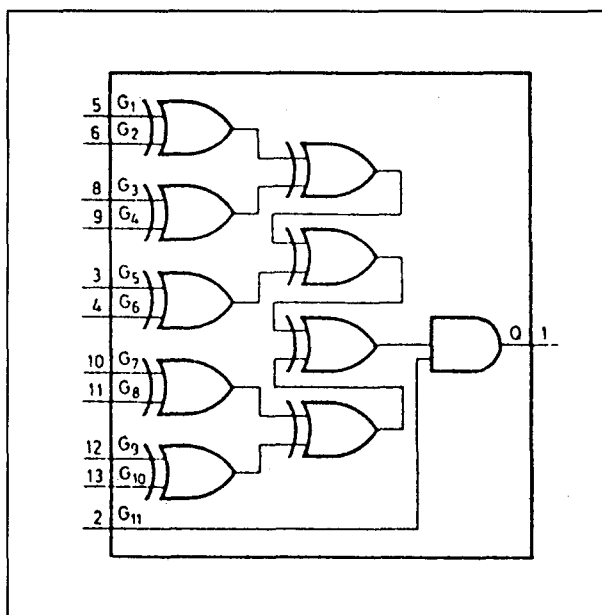
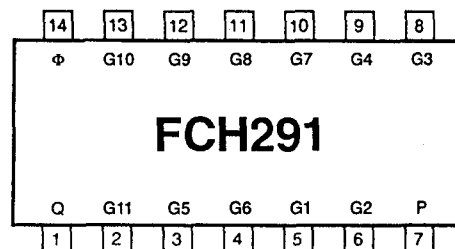
7.3 Diverse schakelingen uit de FC-serie DTL

FCH 291

enkele 10 bit parity-checker

De FCH291 bevat 9 EXOR-functies, gevolgd door een AND-poort. Wanneer een oneven aantal ingangen (G1 tot en met G10) HOOG is, zal de uitgang HOOG zijn (als G11 tenminste ook HOOG is). In de andere gevallen zal de uitgang LAAG zijn. (fan-out = 7)

Figuur 6/7.3-291.

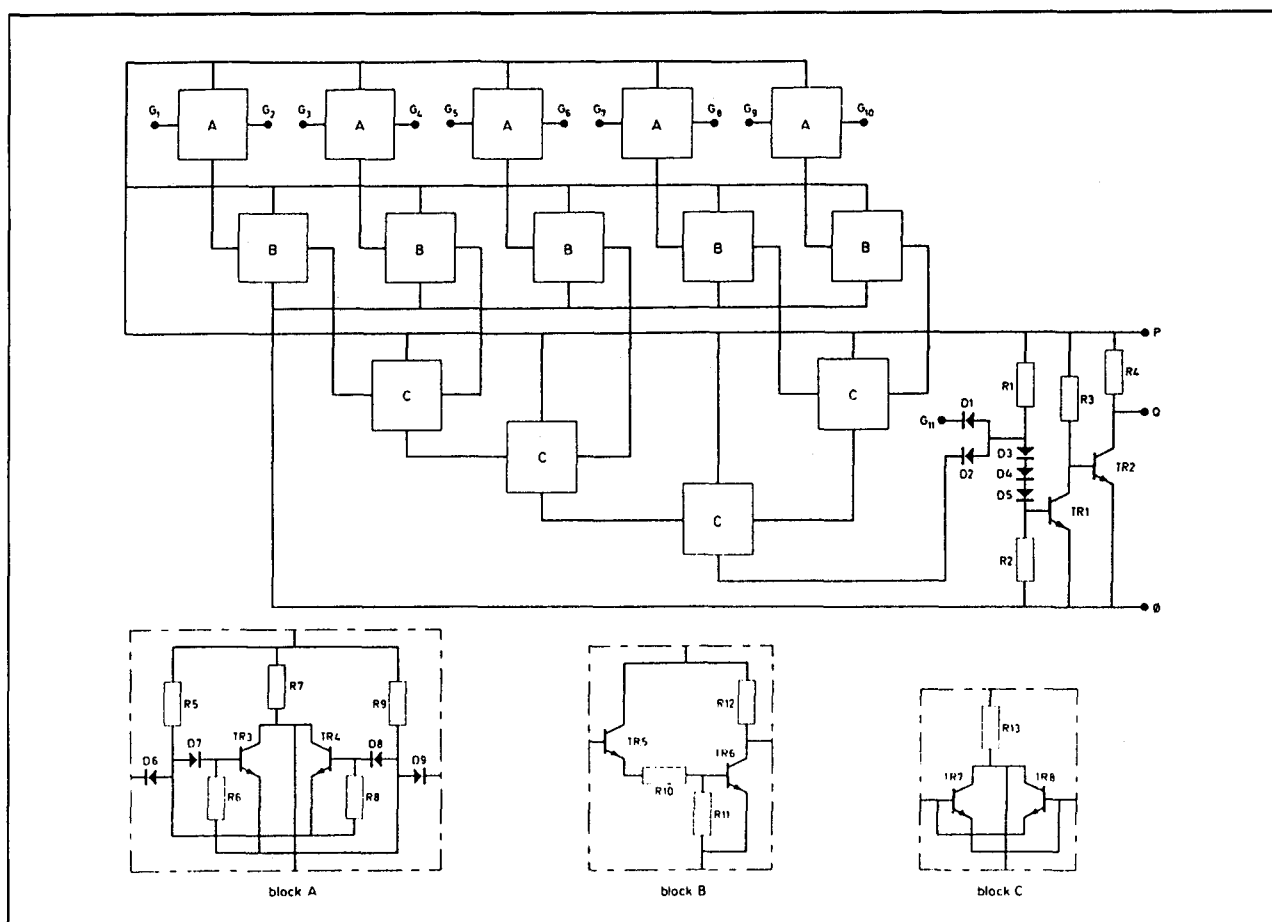


Logisch symbool.

7.3 Diverse schakelingen uit de FC-serie DTL

G ₁	G ₂	G ₃	G ₄	G ₅	G ₆	G ₇	G ₈	G ₉	G ₁₀	G ₁₁	Q
Even number of inputs HIGH										H	L
Odd number of inputs HIGH										H	H
X	X	X	X	X	X	X	X	X	X	L	L

Waarheidstabel.



Logisch schema.